

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра автоматизованих систем обробки інформації та управління

УДК 004.067

«До захисту допущено»

В.о. завідувача кафедри

О.А.Павлов
(ініціали, прізвище)

“ ” 2019 р.

Дипломний проект
на здобуття ступеня бакалавра

з напрямку підготовки 6.050101 «Комп'ютерні науки»

на тему: *Комплекс задач з моніторингу фінансових показників
бюджетної організації*

Виконала: студентка 4 курсу, групи ІС-51

Прокопенко Ірина Вікторівна

(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц., к.т.н., доц. Гриша О.В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

**Консультант з
графічної
документації**

ст. викл. Москаленко Н.В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студентка

(підпис)

Київ – 2019 року

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. Схема структурна діяльності

2. Схема структурна варіантів використання

3. Схема бази даних

4. Схема структурна класів програмного забезпечення

5. Схема структурна послідовності

6. Схема структурна компонентів програмного забезпечення

7. Рішення з математичного забезпечення

8. Креслення вигляду екранних форм

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «15» лютого 2019 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення рекомендованої літератури	24.02.2019	
2.	Аналіз існуючих методів розв'язання задачі	27.03.2019	
3.	Постановка та формалізація задачі	30.03.2019	
4.	Розробка інформаційного забезпечення	02.04.2019	
5.	Алгоритмізація задачі	07.04.2019	
6.	Обґрунтування використовуваних технічних засобів	10.05.2019	
7.	Розробка програмного забезпечення	12.05.2019	
8.	Налагодження програми	15.05.2019	
9.	Виконання графічних документів	17.05.2019	
10.	Оформлення пояснювальної записки	18.05.2019	
11.	Подання ДП на попередній захист	30.05.2019	
12.	Подання ДП на основний захист	03.06.2019	
13.	Подання ДП рецензенту	05.06.2019	

Студент

_____ І.В. Прокопенко
(підпис)

Керівник проекту

_____ О.В. Гриша
(підпис)

[illegible]

Пояснювальна записка до дипломного проекту

на тему: Комплекс задач з моніторингу фінансових показників
бюджетної організації

Київ – 2019 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з п'яти розділів, містить 19 рисунків, 14 таблиць, 1 додатку, 16 джерел.

Дипломний проект присвячений розробці комплексу задач з моніторингу фінансових показників бюджетних організації, тобто оптимізації продукту MASTER шляхом розробки функціоналу «Монітор керівника» для моніторингу, прогнозування фінансових показників та швидкого реагування на зміни станів фінансових показників.

Для розуміння загальних положень було описано процеси діяльності, варіанти використання, описано та проаналізовано існуючі аналоги, та коректно поставлено задачу із визначеними метою та задачами.

Для уявлення про інформаційне забезпечення було визначено вхідні та вихідні дані, було створено та описано базу даних.

Математичному забезпеченню присвячений пошуку найкращого методу прогнозування.

Програмне забезпечення забезпечило описанням засоби розробки, висуванням вимог до технічного забезпечення та вибором архітектуру програмного забезпечення.

Технологічний розділ надав змогу описати інструкція користувача та провести тестування комплексу задач.

ФІНАНСОВІ ПОКАЗНИКИ, ПРОГНОЗ, МОНІТОРИНГ, МОНІТОР КЕРІВНИКА, ПРОДУКТ MASTER.

					ДП ІС-5121.1181-с.ПЗ				
		Прізвище	Підпис	Дата					
Розроб.		Прокопенко І.В.			Комплекс задач з моніторингу фінансових показників бюджетних організацій	Літ.	Лист	Листів	
Перевірів.		Гриша О.В.					2	65	
Н. кон.		Москаленко Н.В.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51			
Затв.		Павлов О.А.							

ABSTRACT

Structure and scope of work. The explanatory note of the diploma project consists of five sections, containing 19 figures, 14 tables, 1 application and 16 sources.

The diploma project is devoted to the development of a set of tasks for monitoring the financial indicators of budget organizations that is, optimization of the product MASTER by developing a functional "Head monitor" for monitoring, forecasting financial indicators and rapid response to changes in financial indicators.

In order to understand the general provisions, processes of activities, variants of use, existing analogues were described and analyzed, and the task with the defined goals and tasks was correctly implemented.

In order to provide information about the information support, input and output data were identified, a database was created and described.

Mathematical security is dedicated to finding the best forecasting method.

The software provided a description of the development tools, the requirements for technical support, and the choice of software architecture.

The technology section provided an opportunity to describe the user's manual and to test a set of tasks.

FINANCIAL INDICATORS, FORECAST, MONITORING, HEAD MONITOR, PRODUCT MASTER.

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКРОРОЧЕНЬ І ТЕРМІНІВ.....	6
ВСТУП	8
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	10
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА	10
1.1.1 <i>Опис процесу діяльності.....</i>	<i>12</i>
1.1.2 <i>Опис функціональної моделі.....</i>	<i>12</i>
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ	13
1.3 ПОСТАНОВКА ЗАДАЧІ.....	18
1.3.1 <i>Призначення розробки.....</i>	<i>18</i>
1.3.2 <i>Цілі та задачі розробки</i>	<i>19</i>
Висновок до розділу	19
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	20
2.1 ВХІДНІ ДАНІ	20
2.2 ВИХІДНІ ДАНІ.....	20
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ	22
Висновок до розділу	23
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	24
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ	24
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ	25
3.3 ОБГРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ.....	25
3.4 ОПИС МЕТОДІВ РОЗВ'ЯЗАННЯ	28
Висновок до розділу	31
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	32
4.1 ЗАСОБИ РОЗРОБКИ	32
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	34
4.2.1 <i>Загальні вимоги</i>	<i>34</i>

4.3	АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	35
4.3.1	Діаграма класів	35
4.3.2	Діаграма послідовності	36
4.3.3	Діаграма компонентів	37
4.3.4	Специфікація функцій	38
	Висновок до розділу	41
5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ	43
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА	43
5.2	ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	49
5.2.1	Мета випробувань	49
5.2.2	Загальні положення.....	49
5.2.3	Результати випробувань	50
	Висновок до розділу	55
	ЗАГАЛЬНІ ВИСНОВКИ	56
	ПЕРЕЛІК ПОСИЛАНЬ	57
	ДОДАТОК А	59

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Активи – ресурси, що контролюються підприємством у результаті минулих подій, використання яких, як очікується, приведе до отримання економічних вигод у майбутньому.

Баланс підприємства – відображення фінансового стану підприємства на конкретну дату.

Бухгалтерський баланс – звіт, який відображає фінансовий стан підприємства на певну дату. Баланс складається з двох частин: активу та пасиву.

Бухгалтерський облік – процес виявлення, вимірювання, реєстрації, накопичення, узагальнення, зберігання та передача інформації про діяльність підприємства зовнішнім або внутрішнім користувачам для прийняття рішень.

Бюджетні організації (установи) – створені у встановленому порядку органами державної влади чи органами місцевого самоврядування неприбуткові заклади, що повністю утримуються за рахунок державного чи місцевого бюджету.

Грошові кошти – готівка, кошти на рахунках у банках та депозити до запитання.

Консалтинг – діяльність з консультування керівників, управлінців з широкого кола питань у фінансовій, комерційній, юридичній, технологічній, технічній, експертній діяльності.

Малий бізнес - це деяка підприємницька діяльність, сформована на власний ризик, що не суперечить чинному законодавству, створює невелику кількість робочих місць. Середньооблікова кількість працюючих за звітний рік не перевищує 50 осіб, а обсяг валового доходу за рік не перевищує 70млн.грн.

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

Моніторинг – це неперервне або періодичне спостереження за станом ринку, щоб оцінити його, вивчити тенденції, досліджувати конкурентів.

Оборотно-сальдова відомість – один із основних бухгалтерських документів, що містить залишки на початок і на кінець періоду та обороти по дебету і кредиту за заданий період для кожного рахунку, підрахунку.

Пасиви – сукупність усіх джерел формування коштів підприємства.

Продукт MASTER – українське рішення для ведення обліку на підприємствах малого, середнього бізнесу та в державних установах.

Розрахунковий рахунок - обліковий запис, що використовується банком або іншою розрахунковою установою для обліку грошових операцій клієнтів. Поточний стан розрахункового рахунку, як правило, відповідає сумі коштів, що належать клієнту.

Середній бізнес – бізнес-термін, яким позначають дещо середнє між малим та великим бізнесами. Середньооблікова кількість працюючих за звітний рік більше 50 осіб, але не перевищує 250, а обсяг валового доходу за рік коливається від 1 млн.грн. до 10 млн.грн.

Фінансові показники або показники фінансового стану підприємства – система показників, що використовується для аналізу фінансового стану підприємства.

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

У травні 2017 року згідно з указом президента України №133/2017, було введено санкції проти ряду російських компаній [5]. Після такого рішення усі бюджетні установи мали відмовитись від використання програми для бухгалтерського обліку в Україні «1С». Та вже у вересні 2017 року на ринку з'являється українське рішення для ведення обліку на підприємствах малого, середнього бізнесу та в державних установах — MASTER [2]. Для бюджетних установ це було просто порятунком, бо продукт відповідає вимогам чинного законодавства України та є адаптованим до українського ринку [1].

Та світ розвивається так швидко, що потрібно прикласти багато зусиль, щоб лишитися на плаву. Особливо нелегко невеликим підприємствам та бюджетним установам.

Саме тому було вирішено вдосконалити продукт MASTER шляхом створення нового функціоналу «Монітор керівника» для можливості візуалізації станів фінансових показників таких, як стан грошових коштів в касі, стан коштів на розрахункових рахунках в банках та доходи і витрати підприємства. Таке відображення фінансових показників значно полегшило б керівнику підприємства, який користуються продуктом MASTER, процес обліку. Але облік це не тільки інформація про діяльність підприємства, це й інформація, яка може бути використана управлінською ланкою для планування певного порядку дій для забезпечення тривалого існування підприємства. Та людина не може точно спрогнозувати, що буде, наприклад, завтра. Саме тому функціонал «Монітор користувача» надає можливість спрогнозувати стани грошових коштів в касі, стани коштів на розрахункових рахунках в банках, стани доходів та витрат підприємства. Отримані прогнози відображатимуться в зручному для розуміння вигляді – у вигляді графіків, які будуть продовженням фактичних даних. За допомогою отриманих простих відображень

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

станів фінансових показників, керівнику стане набагато легше спланувати подальше керування та забезпечити отримання стабільних доходів та стійкого фінансового становища підприємства.

Таким чином додатковий функціонал продукту MASTER «Монітор керівника» стане особистим помічником керівника підприємства з питань обліку та планування. «Монітор користувача» допоможе швидше реагувати на зміни економічного становища тому, що якісна картинка станів коштів в касі, станів коштів на розрахункових рахунках в банках, станів доходів та витрат підприємства дає змогу легко та просто аналізувати те, що відбувається всередині підприємства.

Дипломний проект присвячений розробці комплексу задач з моніторингу фінансових показників бюджетних організацій.

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

В сучасному світі спостерігається дуже швидка зміна подій. Такі зміни можуть не найкращим чином вплинути на невелике підприємство. Тому таким підприємствам потрібно якимось підлаштовуватись під ці зміни та максимально швидко реагувати на них.

Одними із найважливіших показників, які характеризують будь-яке підприємство – кошти, які має підприємство. Саме тому, впевненість підприємства у завтрашньому дні залежить від суми коштів, які будуть в цього підприємства завтра. Кошти можуть бути у вигляді готівки в касі та на розрахункових рахунках в банках.

Продукт MASTER – це українське рішення для ведення обліку на підприємствах малого, середнього бізнесу та в державних установах. Цей продукт забезпечує своїх користувачів можливістю ведення обліку підприємства [2].

Саме для полегшення ведення обліку і створюється «Монітор керівника» як додатковий функціонал MASTER, який відобразить всю інформацію про такі фінансові показники:

- наявність грошових коштів в касі за певний період;
- наявність коштів на розрахункових рахунках в банках за певний період;
- баланс підприємства за певний період.

Наявність грошових коштів в касі – це кількість обігових коштів, які знаходяться на кінець дня в касі.

Наявність коштів на розрахункових рахунках в банках – це кількість обігових коштів, які знаходяться на кінець дня на розрахункових рахунках в банках.

Баланс підприємства – це фінансовий стан підприємства в грошовому вираженні на певну дату, що виражається у двох розрізах:

- за складом, тобто актив (ресурси, контрольовані підприємством у результаті минулих подій, використання яких, як очікується, приведе до отримання економічних вигод у майбутньому);
- за джерелами формування, тобто пасив (сукупність всіх джерел формування коштів підприємства).

Такий систематичний контроль наведених вище фінансових показників представлятиметься в зручному для розуміння вигляді – у вигляді графіків. Такий метод візуалізації полегшить роботу керівнику підприємства. Він допоможе побачити загальну картину стану фінансових показників, швидкість зміни фінансових показників. Така картина допоможе швидше проаналізувати стани не тільки фінансових показників, а стан підприємства загалом, що допоможе швидко зробити висновок щодо того, які заходи необхідно вжити, щоб підприємство надалі існувало та успішно розвивалось.

Щоб переглянути інформацію про залишки на початок і на кінець періоду, та обороти по дебету та по кредиту за вказаний період для кожного рахунку, функціонал надає змогу сформувати оборотно-сальдову відомість кожного фінансового показнику окремо.

Наприклад, оборотно-сальдова відомість по динаміці на розрахункових рахунках детально відобразить інформацію про залишки та на якому рахунку вони зберігаються [3].

Аналогічно оборотно-сальдову відомість можна будувати і по динаміці по касі.

З оборотно-сальдової відомості формується бухгалтерський баланс шляхом розрахунку сальдо по бухгалтерських рахунках і перенесення їх в

сам баланс. Оборотно-сальдові відомості використовуються для перевірки бухгалтерських записів на наявність арифметичних помилок [3].

1.1.1 Опис процесу діяльності

Об'єктом оптимізації продукту MASTER є процес прогнозування стану коштів в касі за певний період, стану коштів на розрахункових рахунках в банках та стану коштів балансу підприємства.

Система призначена для моніторингу фінансових показників у зручній для розуміння формі.

Маючи придбаним продукт MASTER, керівник може користуватись функціоналом моніторингу фінансових показників «Монітор керівника». Використовуючи ті дані, які зберігаються у базі даних замовника (тобто базі даних підприємства) та ті дані, які додаються до бази даних кожного дня, система сама відображає стан фінансових показників за останній тиждень діяльності підприємства. Тобто при кожному вході до інтерфейсу «Монітор керівника», будуть відображатися стани коштів в касі, стани коштів на розрахункових рахунках в банках та баланс підприємства за останній тиждень.

Щоб мати змогу спрогнозувати майбутній стан фінансових показників, керівник має вибрати режим прогнозування, тобто поставити галочку, та ввести кінцеву дату для прогнозу. На введений період створиться прогноз, який буде відображено поруч з фактичними станами фінансових показників, але він буде відображатись іншим кольором.

Схема структурна діяльності наведена в графічному матеріалі.

1.1.2 Опис функціональної моделі

Актором системи є: керівник підприємства.

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Визначимо, які функції він виконує в системі, для цього наведемо таблицю 1.1, в якій описаний актор, функції та їх описи.

Таблиця 1.1 – Опис функцій системи

Актор	Функція	Опис функції
Керів- ник	1. Введення періоду моніторингу	Керівник має змогу вказати потрібний йому період моніторингу
	2. Перегляд графіку фінансового показника	Керівник має змогу переглянути графіки станів фінансових показників банку, каси та аналізу заборгованостей.
	3. Введення періоду для прогнозування	Керівник має змогу сформувати прогноз та вказати кінцеву дату прогнозу.
	4. Перегляд спрогнозованих даних	Керівник має змогу переглянути спрогнозований стан фінансових показників.
	5. Перехід до оборотно-сальдової відомості	Керівник має змогу перейти до оборотно-сальдової відомості від обраного фінансового показника.

Схема структурна варіантів використання представлена відповідно до визначених функцій та наведена в графічному матеріалі.

1.2 Огляд наявних аналогів

Існує багато різного програмного забезпечення для обліку підприємств. Розглянемо деякі з них.

«SAP SE» – німецька корпорація - розробник програмного забезпечення та надавач послуг консалтингу, яка виробляє корпоративне програмне за-

безпечення та забезпечує підтримку програм для компаній будь-якого розміру в усьому світі. Один із модулів SAP SE – модуль PP відповідає за планування потреб в матеріалах, формування планових і технологічних замовлень, оперативне ведення витрат матеріалів на технологічні замовлення, облік виробництв, формування оперативних розрахунків [4]. Вигляд застосунку «SAP SE» представлений на рисунку 1.3.

Продукт MASTER потребує в можливості моніторингу та прогнозування саме фінансових показників, а SAP SE не надає такої можливості.

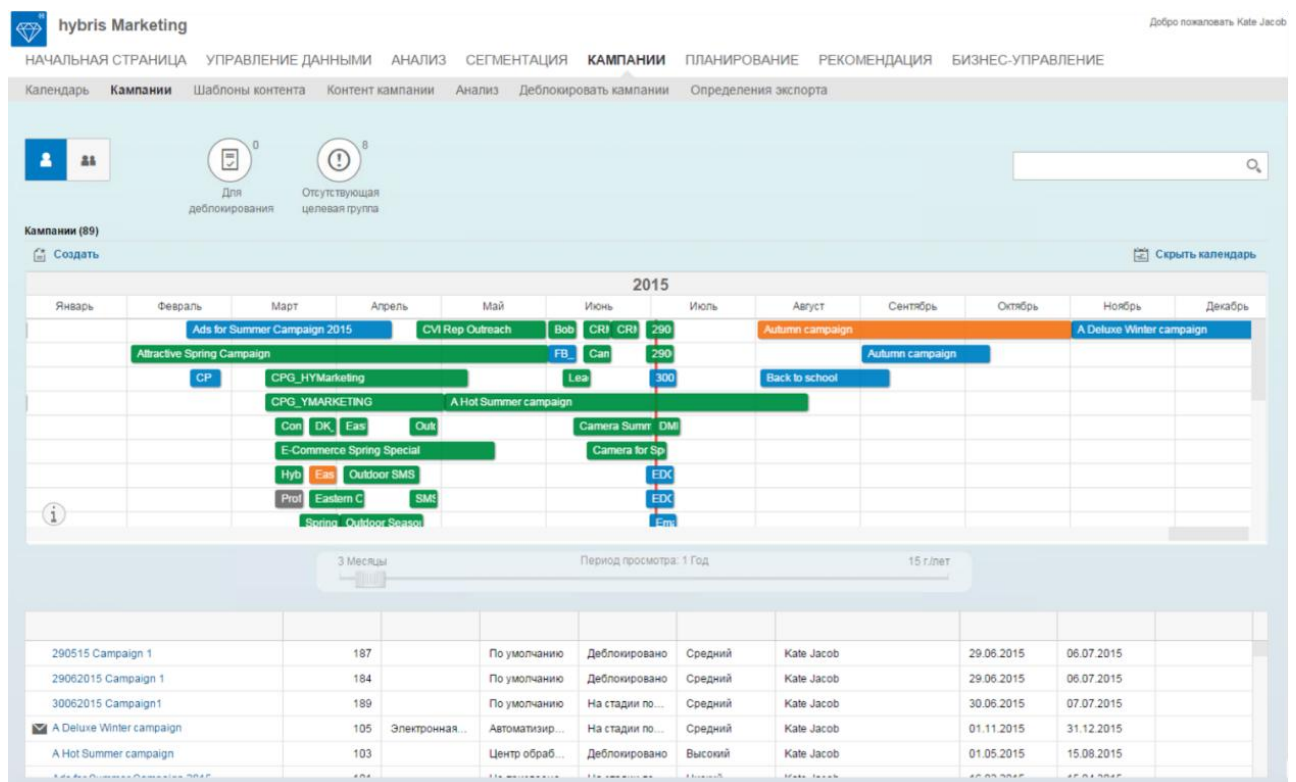


Рисунок 1.3 – Интерфейс «SAP SE»

«ІС: Бухгалтерія для України» – це програмний продукт для ведення бухгалтерського і податкового обліку в комерційних організаціях, а також для формування стандартної бухгалтерської, статистичної та податкової звітності. Основними можливостями продукту є:

- ведення обліку декількох організацій в єдиній інформаційній базі;

- облік «від документу», застосування типових операцій;
- складський облік;
- облік торговельних операцій;
- облік комісійної торгівлі;
- облік операцій з тарою;
- облік банківських і касових операцій;
- облік розрахунків з контрагентами;
- облік основних засобів, нематеріальних і малоцінних активів;
- облік основного й допоміжного виробництва;
- облік непрямих витрат;
- облік напівфабрикатів;
- облік ПДВ;
- облік заробітної плати, кадровий облік;
- підтримка різних схем оподаткування;
- податковий облік з податку на прибуток;
- спрощена система оподаткування;
- завершальні операції періоду;
- стандартні бухгалтерські звіти [5].

Вигляд застосунку «1С: Бухгалтерія для України» представлений на рисунку 1.4.

1С є забороненим в Україні з 2017 року згідно з указом президента України №133/2017 Про рішення Ради національної безпеки і оборони України від 28 квітня 2017 року «Про застосування персональних спеціальних економічних та інших обмежувальних заходів (санкцій)» [6]. До того ж 1С не має можливості прогнозування.

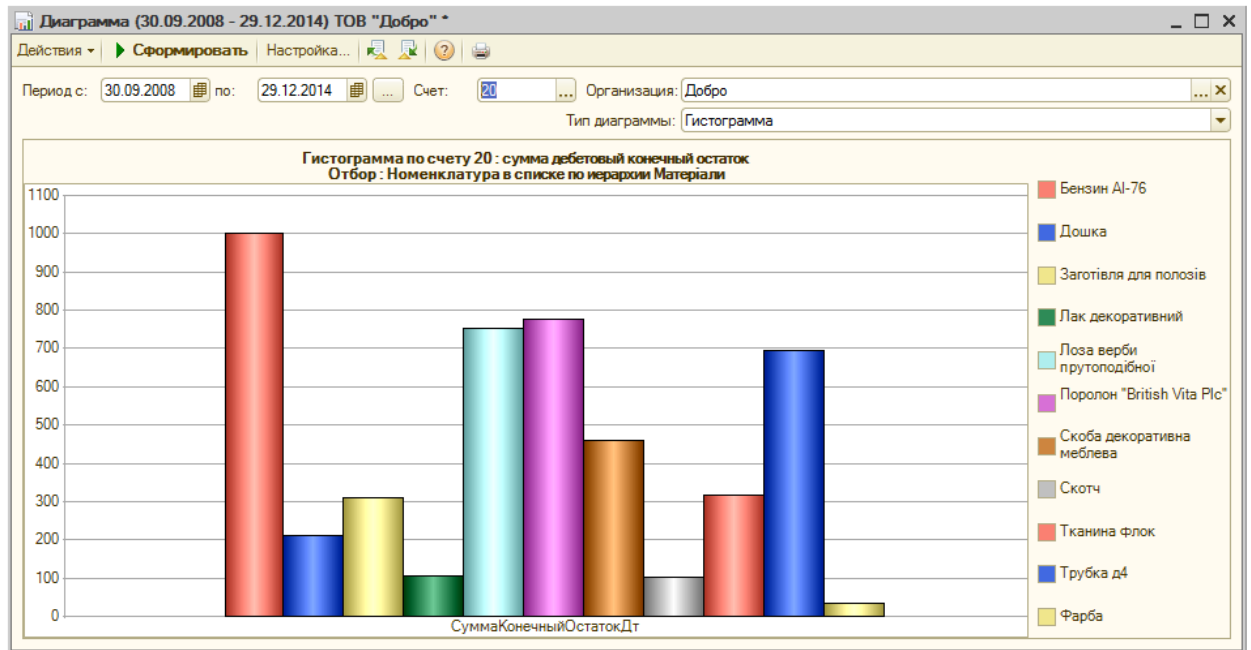


Рисунок 1.4 – Інтерфейс «1С: Бухгалтерія для України»

«**Novo Forecast Enterprise**» – надійна, продуктивна і проста у використанні платформа для автоматизації прогнозування та спільного планування. Дана платформа автоматизує 5 трудомістких процедур підготовки даних для прогнозування і перетворює «сирі» дані з багатьох джерел на масиви, придатні для розрахунку прогнозів високої точності. Novo Forecast автоматично аналізує тимчасові ряди, підбирає модель прогнозування і розраховує прогноз на будь-якому рівні деталізації - за географічними локаціями, клієнтам, товарними категоріями, товарам, місяцях, днях. Для підбору моделей використовуються алгоритми машинного навчання і бібліотека з 3000+ комбінацій прогнозних моделей [7]. Вигляд застосунку «Novo Forecast Enterprise» представлений на рисунку 1.5.

Так як продукт MASTER використовують невеликі підприємства, то даних для використання алгоритмів машинного навчання може не вистачити і тоді прогноз буде хибним. Саме тому в даному випадку не доцільно використовувати прогнозування Novo Forecast Enterprise.

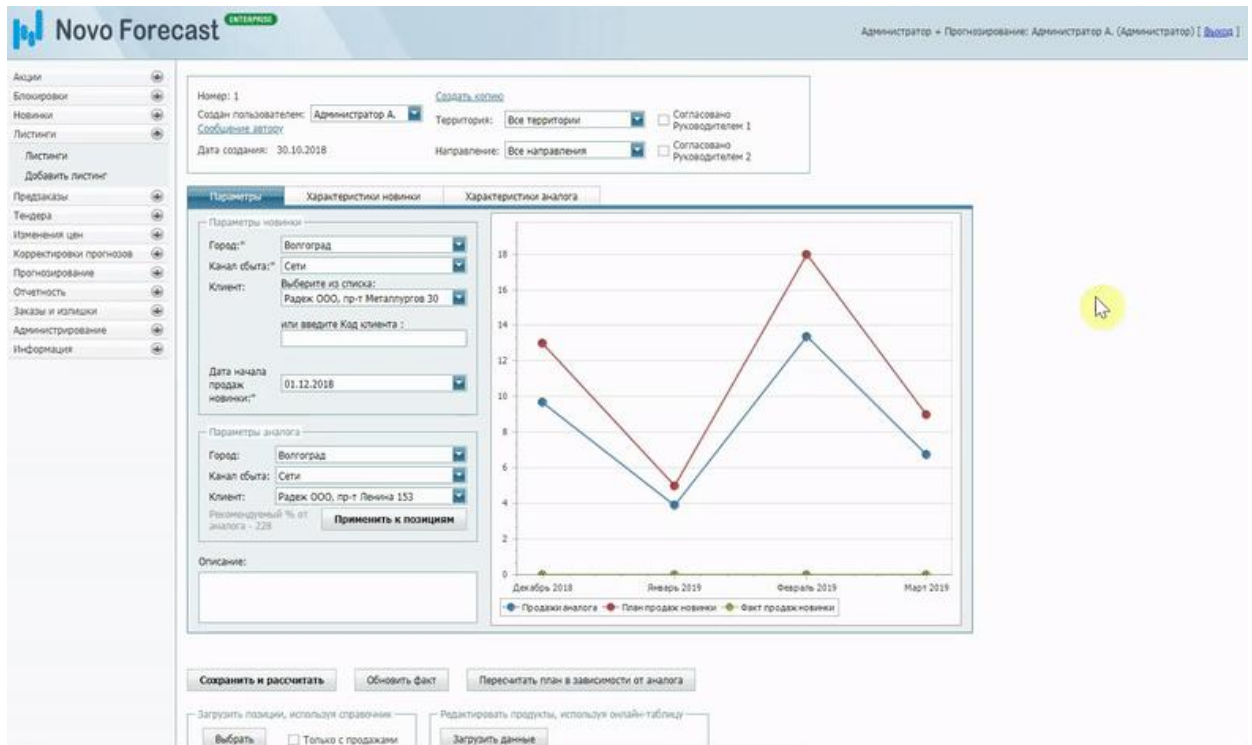


Рисунок 1.5 – Интерфейс «Novo Forecast Enterprise»

«OMP Plus» - система призначена для планування ланцюжків поставчань. OMP Plus забезпечує підвищення точності прогнозів за рахунок спільного процесу прогнозування, тобто можливість спільної роботи з інформацією про попит. Під час розрахунку прогнозу система враховує зміну попиту, ціни, обсягів виробництва, вплив спеціальних акцій і багато іншого. Крім того система управляє життєвим циклом продуктів, аналізує рівні страхового запасу і враховує останні тенденції ринкового попиту [8]. Вигляд застосунку «OMP Plus» представлений на рисунку 1.6.

На жаль, перевести підприємство за один день з однієї системи планування на іншу – неможливо. Тому простіше і дешевше розробити додатковий функціонал для можливості прогнозування в продукті MASTER.

Змн.	Арк.	№ докум.	Підпис	Дата



Рисунок 1.6 – Інтерфейс «OMP Plus»

Через те, що більшість продуктів для обліку підприємства, або не мають можливості моніторингу та прогнозування фінансових показників, або розраховані на великі підприємства, то оптимізація продукту MASTER шляхом реалізації функціоналу «Монітор керівника» є доцільною та актуальною.

1.3 Постановка задачі

1.3.1 Призначення розробки

Призначенням розробки даного функціоналу є вдосконалення продукту MASTER шляхом розроблення функціоналу «Монітор керівника» для можливості моніторингу та прогнозування таких фінансових показників, як стан грошових коштів в касі, стан коштів на розрахункових рахунках в банках, стан балансу підприємства, задля можливості передбачення змін економічної ситуації та для полегшення роботи керівника підприємства щодо планування кращого функціонування цього підприємства.

1.3.2 Цілі та задачі розробки

Метою розробки є пришвидшення процесу сприйняття інформації щодо змін станів фінансових показників, що дає змогу спрогнозувати стан фінансових показників та миттєво зреагувати на ці зміни за рахунок візуалізації даних.

Для досягнення поставлених цілей необхідно розв'язати наступні задачі:

- відобразити фактичний стан фінансових показників, таких як наявність грошових коштів в касі, наявність коштів на розрахункових рахунках в банках та баланс підприємства, на графіках;
- спрогнозувати стан фінансових показників;
- відобразити прогноз фінансових показників, таких як наявність грошових коштів в касі, наявність коштів на розрахункових рахунках в банках та баланс підприємства, на графіках, відобразивши їх іншим кольором поруч з фактичними даними;
- надати можливість переходити до оборотно-сальдової відомості з будь-якого фінансового показнику.

Висновок до розділу

В цьому розділі було обґрунтовано доцільність створення додаткового функціоналу продукту MASTER – «Монітор керівника» як засобу для моніторингу фінансових показників, таких як наявність грошових коштів в касі, наявність коштів на розрахункових рахунках в банках та баланс підприємства. Було описано предметне середовище, шляхом опису процесу діяльності та функціональної моделі. Було знайдено аналоги та виконано порівняння запропонованої ідеї прогнозу в інших схожих системах. Було визначено цілі та задачі розробки функціоналу «Монітор керівника».

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідними даними для системи є:

- інформація про наявність грошових коштів в касі:
 - 1) дата,
 - 2) кількість коштів в касі;
- інформація про наявність коштів на розрахункових рахунках в банку:
 - 1) дата,
 - 2) кількість коштів на розрахункових рахунках в банках;
- інформація про доходи та витрати підприємства:
 - 1) дата,
 - 2) кількість коштів доходів підприємства,
 - 3) кількість коштів витрат підприємства.

2.2 Вихідні дані

Вихідними даними системи є:

- дані для моніторингу:
 - 1) відображення стану грошових коштів в касі за вказаний період (макет екранної форми наведено на рисунку 2.1),
 - 2) відображення стану коштів на розрахункових рахунках в банку за вказаний період (макет екранної форми наведено на рисунку 2.2),

3) відображення стану балансу підприємства, доходи та витрати підприємства за вказаний період (макет екранної форми наведено на рисунку 2.3);

– дані для прогнозу:

- 1) відображення прогнозованого стану грошових коштів в касі за вказаний період,
- 2) відображення прогнозованого стану коштів на розрахункових рахунках в банку за вказаний період,
- 3) відображення стану балансу підприємства, тобто прогноз доходів та витрат підприємства за вказаний період.



Рисунок 2.1 – Відображення стану грошових коштів в касі



Рисунок 2.2 – Відображення стану коштів на розрахункових рахунках



Рисунок 2.3 – Відображення стану балансу підприємства

2.3 Опис структури бази даних

Для коректної роботи з функціоналом «Монітор керівника» потрібен доступ до бази даних замовника. Але, щоб відтворити сценарій моніторингу та прогнозування фінансових показників, потрібно мати не лише доступ до бази даних замовника, а й можливість коригувати записи в ній. Внесення таких правок може привести до втрати даних або до їх некоректності.

Тому, в системі, на основі якої було створено продукт MASTER, існує модуль, який копіює дані з бази даних замовника та вставляє їх реляційну базу даних MySQL. У цій базі даних є 3 таблиці, перелік яких наведено у таблиці 2.1.

Таблиця 2.1 – Перелік сутностей

№	Назва таблиці	Сутність
1	`Cash`	Наявність грошових коштів в касі
2	`Bank`	Наявність коштів на розрахункових рахунках в банку
3	`Balance`	Баланс підприємства

Саму схему бази даних наведено в графічному матеріалі.

Таблиця 2.2 містить опис структури бази даних.

Таблиця 2.2 – Опис структури БД

Назва таблиці	Назва поля	Опис поля
Cash	date	Дата – день, за який рахується кількість обігових коштів в касі
	cashVal	Кількість грошових коштів в касі
Bank	date	Дата – день, за який рахується кількість обігових коштів на розрахункових рахунках в банках
	bankVal	Кількість коштів на розрахункових рахунках а банках
Balance	guidBalance	Унікальний ідентифікатор балансу
	date	Дата – день, за який рахується кількість обігових коштів в касі
	income	Дохід підприємства
	consumption	Витрати підприємства

Таблиці бази даних ніяк не зв'язані між собою тому, що дані, які надходять до них є незалежними, тобто зміна даних в одній таблиці не призводить до зміни даних в іншій.

Висновок до розділу

В даному розділі було описано вхідні, які знаходяться за допомогою модуля-імпортеру з бази даних замовника та вихідні дані, які будуть зберігатись в локальних таблицях і будуть доступні лише при поточній роботі з функціоналом. Саме тому, було створено та описано структуру бази даних та обґрунтовано, чому база даних має саме такий вигляд.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Функціонал, що розробляється повинен повністю задовольняти потреби націленого користувача. В даному випадку роль такого користувача відіграє керівник підприємства, яким було придбано продукт MASTER – програмний продукт для ведення комплексного обліку [1].

«Монітор керівника» - засіб для досягнення зразу двох цілей. По-перше, керівник має змогу переглянути стан фінансових показників таких, як стан коштів в касі, стан коштів на розрахункових рахунках в банках та баланс підприємства, за вказаний період. Така можливість потрібна для того, щоб наглядно бачити як розвивається підприємства. По-друге, керівник має змогу сформулювати прогноз того, яка кількість коштів буде в касі, яка кількість коштів буде на розрахункових рахунках в банках та який буде баланс підприємства на вказаний період для прогнозування. Також керівник має змогу переглянути уже сформований період для того, щоб по ньому вже зробити висновки про подальшу стратегію підприємства.

До того ж, «Монітор керівника» наділений можливістю переходу до оборотно-сальдової відомості, щоб переглянути інформацію про залишки на початок і на кінець періоду, та обороти по дебету та по кредиту за вказаний період для кожного рахунку [3].

Саме тому, задача такого функціоналу полягає у відображенні поточних станів коштів в каси, коштів на розрахункових рахунках в банках та балансу підприємства, та в відображенні максимально точних прогнозів цих станів.

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

3.2 Математична постановка задачі

Призначенням задачі є відображенні поточних станів коштів в каси, коштів на розрахункових рахунках в банках та балансу підприємства, та в відображенні максимально точних прогнозів цих станів.

На етапі формування відображень фінансових показників наявними є такі дані:

- стан грошових коштів в касі та дата, на яку цей стан було зафіксовано;
- стан коштів на розрахункових рахунках в банках та дата, на яку цей стан було зафіксовано;
- баланс підприємства, тобто значення доходів та витрат за певну дату.

Ціль задачі – маючи на вході вище перераховані дані, відобразити стан фінансових показників та, при необхідності, яка визначається за бажанням керівника, створити такий прогноз, який буде нести максимально точну інформацію щодо майбутніх станів коштів в касі, коштів на розрахункових рахунках в банках та балансу підприємства.

3.3 Обґрунтування методу розв'язання

Поставлена задача прогнозування може бути вирішена за допомогою багатьох методів. Розглянемо декілька з них.

Метод простих ковзаючих середніх – метод, при якому прогнозування відбувається за рахунок усереднення всіх існуючих даних. Увага цього методу прикована більше до останніх даних, тобто як тільки поступають нові дані, то вони зразу ж включаються до усереднення, а старіші – виключаються відповідно. Заново обраховане ковзне середнє використовується для ство-

рення прогнозу на наступний період. Не менш важливим є і той факт, що для методу простих ковзаючих середніх може бути задіяно лише певну фіксована кількість останніх спостережень, а кількість точок, які будуть усереднюватись, не змінюється з часом. Недоліком цього методу є те, що він потребує статистичного обліку і зберігання всіх минулих даних.

Метод «Коріння трави» - метод, який формує прогноз на основі тих даних, що отримуються з самого кореня ієрархії, а потім послідовно підсумовуються на вищих рівнях. Даний метод передбачає те, що індивід, який знаходиться найближче до кінцевого споживача або до кінцевого використання продукції, знає майбутні потреби краще інших. Прогнози нижнього рівня підсумовуються і передаються на наступний більш високий рівень. Процедура буде повторюватися до тих пір, поки не буде досягнуто верхній рівень.

Метод Дельфі – метод, при якому група експертів відповідає на питання. Під час цього методу керівник групи компілює результати і формулює нове питання, яке виноситься на розгляд групі.

Таким чином під час процесу вивчення проблеми не спостерігається групового тиску чи домінування думок окремих експертів.

Основними стадіями методу Дельфі є:

- вибір експертів. Експертами повинні бути професіонали, які представляють різні області;
- за допомогою опитування, отримують прогноз (зауваження або уточнення до прогнозів) від усіх учасників експертних груп;
- підведення підсумків і роздача отриманих висновків учасникам з відповідними новими питаннями;
- знову підводяться висновки шляхом вдосконалення прогнозів та умов. Розроблюються нові питання;

- в разі потреби повторюється попередній пункт та відбувається ознайомлення з результатами всіх учасників.

Метод Хольта – метод 2-параметричного експонційного злагодження з урахуванням тренду. Суть алгоритму полягає в тому, що значення рівня та тренду злагоджуються за допомогою експонційного злагодження. У цьому методі враховується локальний лінійний тренд, присутній у тимчасових рядах. Якщо у тимчасових рядах є тенденція до росту, то разом з оцінкою поточного рівня необхідна і оцінка нахилу. У даній методиці значення рівня і нахилу згладжуються безпосередньо, при цьому використовуються різні постійні згладжування для кожного з них. Ці постійні згладжування дозволяють оцінити поточний рівень і нахил, уточнюючи їх всякий раз, коли з'являються нові спостереження.

Для знаходження прогнозу методом Хольта, необхідно вирішити 3 рівняння, за допомогою яких буде знайдено експоненційно злагоджений ряд або оцінку поточного рівню, оцінку тренду та безпосередньо прогноз на потрібну кількість періодів вперед.

Метод Вінтерса – метод 3-параметричного експонційного злагодження з урахуванням тренду та сезонних варіацій. На відміну від методу Хольта, метод Вінтерса дозволяє враховувати сезонні коливання при прогнозуванні, тобто метод може «бачити» в передісторії. При цьому робиться спроба відловити сезонні складові в даних.

Модель Вінтерса описується чотирма рівняннями:

- рівняння експоненційно злагодженого ряду - рівняння, яке коригує злагожені ряди шляхом виключення сезонних ефектів, якщо такі мають місце у вхідних даних;
- рівняння оцінки тренду;
- рівняння оцінки сезонності;

- безпосередній прогноз на задану кількість періодів вперед.

Для прогнозування на довгі періоди методи Хольта та Вінтерса не здатні. Тому, було створено **метод Хольта-Вінтерса** - метод 3-параметричного експоненційного згладження з трендом та сезонністю.

Суть методу полягає в прогнозуванні часових рядів, коли в структурі даних наявні сформовані тренд та сезонність. Перевагою методу Хольта-Вінтерса є можливість успішно прогнозувати на середній та довгий періоди. Це можливо тому, що цей спосіб має змогу виявляти мікротренди, тобто тренди, які відносяться до коротких періодів, в момент часу, що безпосередньо передуює прогнозу, і екстраполювати ці тренди в майбутнє.

В функціоналі «Монітор керівника» прогнозування може відбуватись і на довгі періоди, що виключає методи Хольта та Вінтерса. Так як створений прогноз повинен бути максимально точним, то не можна приймати до уваги лише останні дані. Таким чином, метод простих ковзаючих середніх та метод коріння та трав не підходять. Метод Дельфі повністю залежить від людського фактору, що не може гарантувати точності. Тому для поставленої задачі прогнозування буде розв'язано за допомогою методу Хольта-Вінтерса.

3.4 Опис методів розв'язання

Метод прогнозування Хольта-Вінтерса – це 3-параметрична модель прогнозу, яка враховує:

- згладжений експоненційний ряд;
- тренд;
- сезонність.

Саме тому, для обчислення прогнозу необхідно виконати такі кроки:

КРОК 1. Обчислення експоненційно-згладженого ряду.

КРОК 2. Визначення значення тренду.

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

КРОК 3. Оцінка сезонності.

КРОК 4. Формування прогнозу.

Розглянемо детальніше прогнозування за методом Хольта-Вінтерса.

На першому етапі необхідно обчислити експоненційно-злагоджений ряд, тобто знайти значення злагоджених величин на поточний період за формулою 3.1.

$$L_t = k * \frac{Y_t}{S_{t-s}} + (1 - k) * (L_{t-1} + T_{t-1}) \quad (3.1)$$

де L_t – злагоджена величина на поточний період,

k – коефіцієнт злагодженості ряду,

S_{t-s} – коефіцієнт сезонності попереднього періоду,

Y_t – поточне значення ряду,

L_{t-1} – злагоджена величина за попередній період,

T_{t-1} – значення тренду за попередній період.

Далі необхідно визначити тренд поточного періоду за формулою 3.2.

$$T_t = b * (L_t - L_{t-1}) + (1 - b) * T_{t-1} \quad (3.2)$$

де T_t – значення тренду поточного періоду,

b – коефіцієнт злагодженості тренду,

L_t – експоненціально згладжена величина за поточний період,

L_{t-1} – експоненціально згладжена величина за попередній період,

T_{t-1} – значення тренду на попередній період.

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

Використовуючи уже знайдені злагоджені величини поточного періоду, необхідно оцінити коефіцієнти сезонності поточного періоду за формулою 3.3.

$$S_t = q * \frac{Y_t}{L_t} + (1 - q) * S_{t-s} \quad (3.3)$$

де S_t – коефіцієнт сезонності поточного періоду,

q – коефіцієнт злагодженості сезонності,

Y_t – поточне значення ряду,

L_t – злагоджена величина поточного періоду,

S_{t-s} – коефіцієнт сезонності за той самий період, що і в попередньому сезоні.

Останнім кроком є безпосереднє формування прогнозу. Прогноз на p періодів вперед знаходиться за формулою 3.4.

$$\widehat{Y_{t+p}} = (L_t + p * T_t) * S_{t-s+p} \quad (3.4)$$

де $\widehat{Y_{t+p}}$ – прогноз методом Хольта-Вінтерса на p періодів вперед,

L_t – експоненційно злагоджена величина за останній період,

p – порядковий номер періоду, на який робимо прогноз,

T_t – тренд за останній період,

S_{t-s+p} – коефіцієнт сезонності за той же період в останньому сезоні [9].

Рішення з математичного забезпечення наведено в графічному матеріалі.

Висновок до розділу

В даному розділі було сформульовано змістовну та математичну постановки задачі відображення поточних станів коштів в каси, коштів на розрахункових рахунках в банках та балансу підприємства, та відображення максимально точних прогнозів цих станів.

Для вирішення поставленої задачі прогнозування було запропоновано декілька різних методів, але вони не повністю задовольняють умовам задачі, поставленої замовниками.

Таким чином, було обрано метод Хольта-Вінтерса. Описаний метод був представлений у вигляді покрокового алгоритму.

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Оскільки продукт MASTER написаний за допомогою мови програмування C#, версія 5.0, то для створення потрібного функціоналу було використано цю саму мову програмування.

На сьогоднішній день мова програмування C# є однією з найпопулярніших в IT-індустрії. Наразі на C# пишуться найрізноманітніші застосунки: від невеликих десктопних програм до великих веб-порталів і веб-сервісів, які кожного дня обслуговують мільйони користувачів [13].

C# є об'єктно-орієнтованою мовою і в цьому плані багато перейняв у Java та C++. Наприклад, C# підтримує поліморфізм, унаслідування, перевантаження операторів, статичну типізацію. Об'єктно-орієнтований підхід дає змогу вирішити задачі, які потребують масштабних, але в той самий час гнучких та застосунків, що легко розширюються [13].

До того ж, C# активно розвивається тому, що здебільшого підтримується компанією Microsoft. Після кожного оновлення, з'являється все більше цікавих можливостей таких, як, наприклад, лямбди, асинхронні методи і таке інше.

Розглянемо основні переваги C#:

- об'єктно-орієнтована мова, тому не потрібно тримати в голові всю інформацію про абстрактні конструкції, що створюються, а працювати по принципу чорного ящика – вхідні дані -> «магія» -> результат;
- наявні синтаксичні готові конструкції, які роблять життя розробника дещо простішим;

- містить багато готових бібліотек та шаблонів, які дозволяють не витрачати час на вигадкування чогось простого;
- гнучка мова програмування. Різноманітність застосунків, які можуть бути створені за допомогою цієї мови необмежена. Це можуть бути застосунки під Windows, мобільні застосунки, веб-застосунки, ігри, застосунки для Android та iOS, які розробляються за допомогою додаткових фреймворків таких, як Xamarin чи Mono.

Таким чином, можна зробити висновок, що мова програмування C# є найвдалішим вибором для розробки побідного функціоналу.

Продукт MASTER містить свою базу даних MsSql.

MsSql – одна з найбільш популярних систем управління базами даних в світі. Така СУБД підходить для різних проектів: від невеликих застосунків до великих високонавантажених проектів.

MsSql має такі характерні особливості:

- потужність. Тобто MsSql працює дуже швидко;
- надійність і безпека, бо MsSql надає шифрування даних;
- простота. З даною СУБД дуже легко працювати та вести адміністрування.

Для організації даних використовується реляційна модель, яка зберігає дані у вигляді таблиць. Кожна таблиця складається з рядків та стовбців. Кожен рядок зберігає окремий об'єкт, а в стовпцях розміщуються атрибути цього об'єкту.

Таким чином, можна зробити висновок, що база даних MsSql для роботи з подібним функціоналом підходить дуже добре, тому її варто використати [14].

Для можливості візуалізації станів фінансових показників було обрано DevExpress Dashboards.

DevExpress – набір інструментів, призначенням якого є створення інтерактивних панелей з нуля та відображення цих панелей на будь-якій з доступних платформ [15].

За допомогою приладової панелі DevExpress Dashboards є можливість створювати проникливі та інформаційно-насичені системи підтримки прийняття рішень для керівників і бізнес-користувачів на різних платформах і пристроях [16].

Для початку необхідно вибрати потрібний елементу інтерфейсу, наприклад графік, навігаційне дерево, прості таблиці та таке інше. Після цього, необхідно закинути дані для відображення на відповідні аргументи, значення та ряди.

DevExpress Dashboards автоматично забезпечує найкращу опцію візуалізації даних. Результати будуть негайними, точними та завжди актуальними. Саме тому використання DevExpress Dashboards є доцільним та обґрунтованим.

4.2 Вимоги до технічного забезпечення

4.2.1 Загальні вимоги

Даний функціонал являє собою оптимізацію продукту MASTER, а саме моніторинг фінансових показників таких, як стан грошових коштів в касі, стан коштів на розрахункових рахунках в банках та баланс підприємства, за вказаний період і призначений для використання керівниками підприємств.

Для роботи з цим функціоналом до складу технічних засобів повинні входити певні компоненти.

Має бути встановлене таке програмне забезпечення:

- операційна система вище Windows 7;

- куплений продукт MASTER або купити продукт з уже розробленим функціоналом (десктопний застосунок або посилання на комплект продукту MASTER).

Комп'ютерна периферія, до складу якої входять:

- монітор;
- мишка або тачпад;
- клавіатура.

4.3 Архітектура програмного забезпечення

4.3.1 Діаграма класів

Діаграма класів – структурна діаграма мови моделювання UML. Діаграма класів описує те, що повинно бути присутнім в системі, що моделюється. Різні компоненти цієї діаграми можуть являти собою класи, які будуть фактично запрограмовані, основні об'єкти або взаємодії між класами та об'єктами.

Діаграма класів надає наступні можливості:

- проілюструвати моделі даних для інформаційних систем, незалежно від їхнього рівня складності;
- краще зрозуміти загальний вигляд схеми застосування;
- візуально висловити будь-які специфічні потреби системи і поширити цю інформацію;
- створити детальні діаграми, які будуть висвітлювати будь-який конкретний код, який необхідно запрограмувати та реалізувати до описаної структури;
- надати незалежний від реалізації опис типів, що використовуються в системі, та пізніше будуть передані між її компонентами [10].

Було створено діаграму, яка описала компоненти, наявні в розроблюваному функціоналі. Такими компонентами є класи дашбордів каси, банку та балансу, які унаслідуються від головного класу дашборду. Також наявний клас, який є інтерфейсом користувача, клас параметрів, який потрібний для розрахунку прогнозу та класи підготування даних до прогнозу і безпосередньо саме прогнозування.

Схема структурна класів програмного забезпечення наведена в графічному матеріалі.

4.3.2 Діаграма послідовності

Діаграма послідовності – різновид діаграми в UML, яка є типом діаграми взаємодії, оскільки вона описує, як і в якому порядку група об'єктів взаємодіє та працює разом. Діаграма використовується для розуміння вимог до нової системи або документування існуючого процесу.

Діаграма послідовності надає наступні можливості:

- вказати деталі випадків використання UML;
- змодельовати логіку складних процедур, функцій або операцій;
- переглянути, як об'єкти та компоненти взаємодіють один з одним задля завершення процесу;
- спланувати та зрозуміти деталізовану функціональність існуючого або майбутнього сценарію [11].

Перед початком роботи з функціоналом «Монітори керівника» об'єкт модуль імпортує дані з бази даних користувача до нашої бази даних.

Сам процес роботи керівника з функціоналом «Монітори керівника» відбувається наступним чином. Керівник входить до інтерфейсу. Щоб бачити якусь інформацію при першому вході до інтерфейсу, об'єкт інтерфейсу посилає запит на базу даних, яка в свою чергу повертає інтерфейсу ті дані, які по-

трібні для відображення. Об'єкт інтерфейс робить перерахунок та повертає картинку на перегляд користувачу.

Аналогічний алгоритм дій відбувається і при оновленні при зміні періоду та при прогнозуванні. Єдине що, при прогнозуванні, об'єкт інтерфейс виконує формування прогнозу.

Схема структурна послідовності наведена в графічному матеріалі.

4.3.3 Діаграма компонентів

Діаграма компонентів – різновид діаграми в UML. Метою цієї діаграми є показати взаємозв'язок між різними компонентами системи. Найчастіше, в об'єктно-орієнтованому програмуванні компонентна діаграма дозволяє старшому розробнику групувати класи на основі спільної мети так, проект розробки програмного забезпечення розглядатиметься на високому рівні.

Діаграма компонентів надає наступні можливості:

- уявити фізичну структуру системи, що розробляється;
- звернути увагу на компоненти системи та їхні відношення;
- виділити поведінку служб, які мають стосунок до інтерфейсу [12].

Було розроблено діаграму компонентів, яка має наступні компоненти:

- request – запити користувача (керівника);
- data Transfer – передача даних між компонентами;
- CRUD Operations - визначає чотири базові функції управління даними «створення, зчитування, редагування і видалення». Використовується для опису конфігурування користувацького інтерфейсу, що полегшує перегляд, пошук та редагування інформації;
- sqlRequests – надсилання запитів до бази даних.

Схема структурна компонентів програмного забезпечення наведена в графічному матеріалі.

4.3.4 Специфікація функцій

Для предметної області оптимізації продукту MASTER шляхом розроблення функціоналу «Монітор керівника», було розроблено методи для кожного класу функціоналу.

В таблицях 4.1 – 4.6 наведено опис методів основних класів. Опис методів має наступну структуру:

- назви методу;
- опису призначення методу;
- опис параметрів, які метод має на вході, якщо такі наявні;
- опис параметрів, якщо такі наявні.

Таблиця 4.1 – Методи класів LeadBank, LeadCash та LeadBalance

Метод	Опис методу
GetAdditionalDataSources()	Отримати додаткові джерела даних
RefreshData()	Виконати оновлення даних додаткових джерел

Таблиця 4.2 – Методи класу LeaderDataEditor

Метод	Опис методу
When()	Метод, що викликається при завантаженні інтерфейсу

Таблиця 4.3 – Методи класу Accmonp

Метод	Опис методу
ScreenWhen()	Метод, що викликається при відкритті екрану
DatefromValueChanged()	Метод, що спрацьовує при зміні значення дати початку періоду
DatetoValueChanged()	Метод, що спрацьовує при зміні значення дати закінчення періоду
DoForecastValueChanged()	Метод, що спрацьовує при зміні потреби прогнозування

Таблиця 4.4 – Методи класу Forecast

Метод	Опис методу	Параметр	Опис параметру
forecastCreating()	Обчислення прогнозу	endDate, beginDate, data	Кінцева дата періоду прогнозування, початкова дата періоду прогнозування, дані, на основі яких буде здійснюватися прогнозування
Balance-Forecast-Update()	Оновлення значень прогнозів, верхніх та нижніх границь точності по балансу підприємства	forecast, isIncome, data	Прогноз, ознака доходу, дані, на основі яких було здійснено прогнозування

Продовження таблиці 4.4

bankCashForecastUpdate()	Оновлення значень прогнозів, верхніх та нижніх границь точності по динаміці по касі та на розрахункових рахунках	forecast, data	Прогноз, дані, на основі яких було здійснено прогнозування
Create()	Зчитування даних для прогнозування та створення прогнозу	mainDataSource, sign, lastDataEqualsForecast	Основне джерело даних, ознака дашборду, ознака для оновлення значень прогнозів
merge()	З'єднання графіків фактичних даних з графіками прогнозів	data, sign, isIncome	Зчитані дані, ознака дашборду, ознака доходу

Таблиця 4.5 – Методи класу DashBoard

Метод	Опис методу
GetAdditionalDataSources()	Отримати додаткові джерела даних
RefreshData()	Виконати оновлення даних додаткових джерел

Таблиця 4.6 – Методи класу HoltWintersAlgorithm

Метод	Опис методу	Параметр	Опис параметру
ets()	Обчислення експоненційно-злагодженого ряду	data	Дані, потрібні для обчислення експоненційно-злагодженого ряду
trendDefine()	Визначення значення тренду	ets	Експоненційно-злагоджений ряд
seasonalityRate()	Оцінка сезонності	trend, ets	Тренд, експоненційно-злагоджений ряд
ForecastCreate()	Формування прогнозу	seasonality, trend, ets	Сезонність, тренд, експоненційно-злагоджений ряд

Висновок до розділу

У цьому розділі було наведено та описано засоби розробки, які були обрані для функціоналу «Монітор керівника», як засіб моніторингу та прогнозування фінансових показників таких, як стан грошових коштів в касі за заданий період, стан коштів на розрахункових рахунках в банках за заданий період, балансу підприємства за заданий період. Було наведено вимоги до технічного та програмного забезпечення, які необхідні для коректної роботи функціоналу та роботи функціоналу взагалі. Було наведено детальний опис діаграм класів та послідовності, які демонструють класи функціоналу та їх порядок взаємодії зі сховищем даних та керівником, тобто користувачем.

Детальний опис діаграми компонентів продемонстрував розбиття програмної частини функціоналу, що розробляється, на структурні компоненти та їх залежності між собою.

Було наведено специфікацію методів, які були задіяні в функціоналі «Монітор керівника».

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Для того, щоб мати змогу користуватись розробленим функціоналом, потрібно мати купленим продукт MASTER. Якщо ця умова виконана, то можна приступати до роботи з «Монітором керівника».

Після входу до інтерфейсу для перегляду керівнику доступні такі закладки:

- закладка «Параметри», на якій керівник буде вводити період, на який йому необхідно відобразити стан фінансових показників;
- закладка «Динаміка по касі», на якій відображається стан грошових коштів в касі за заданий період;
- закладка «Динаміка на розрахункових рахунках», на якій відображається стан коштів на розрахункових рахунках в банках за заданий період;
- закладка «Дохід», на якій відображаються доходи та витрати підприємства за заданий період, а також баланс підприємства за цей же період.

На рисунку 5.1 зображений вигляд інтерфейсу «Монітор користувача».

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

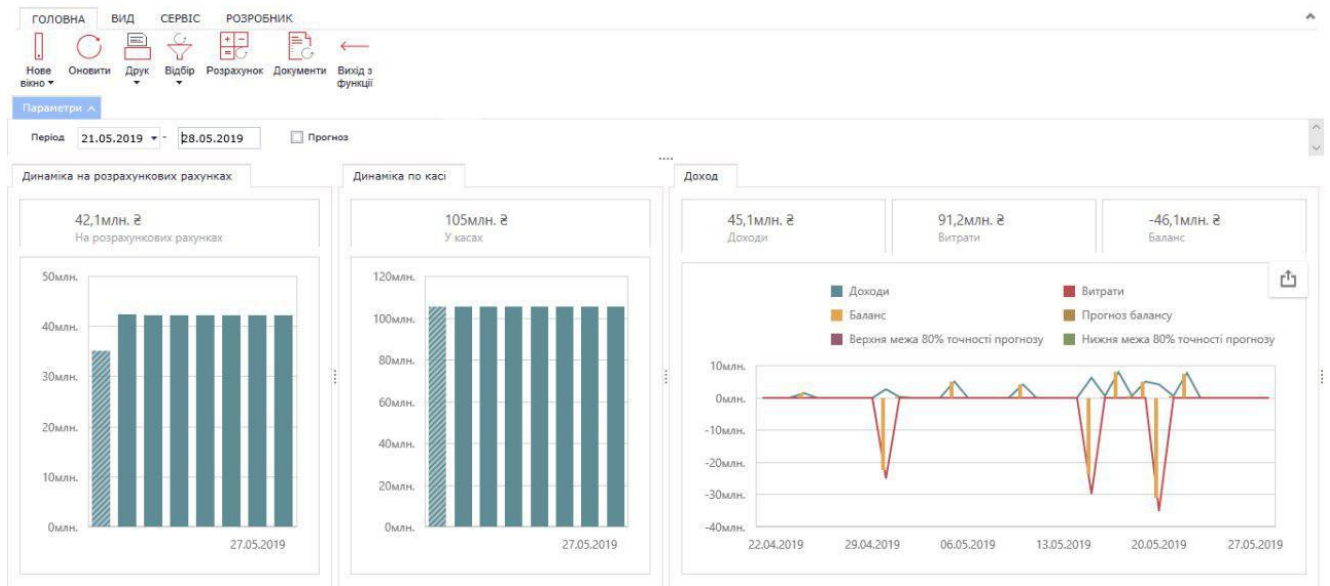


Рисунок 5.1 – Вигляд інтерфейсу «Монітор керівника»

Якщо керівника цікавить стан фінансових показників за інший період, то достатньо змінити період на закладці «Параметри», натиснути на кнопку «Розрахунок» та вибрати «Оновити». Спосіб оновлення даних представлений на рисунку 5.2.

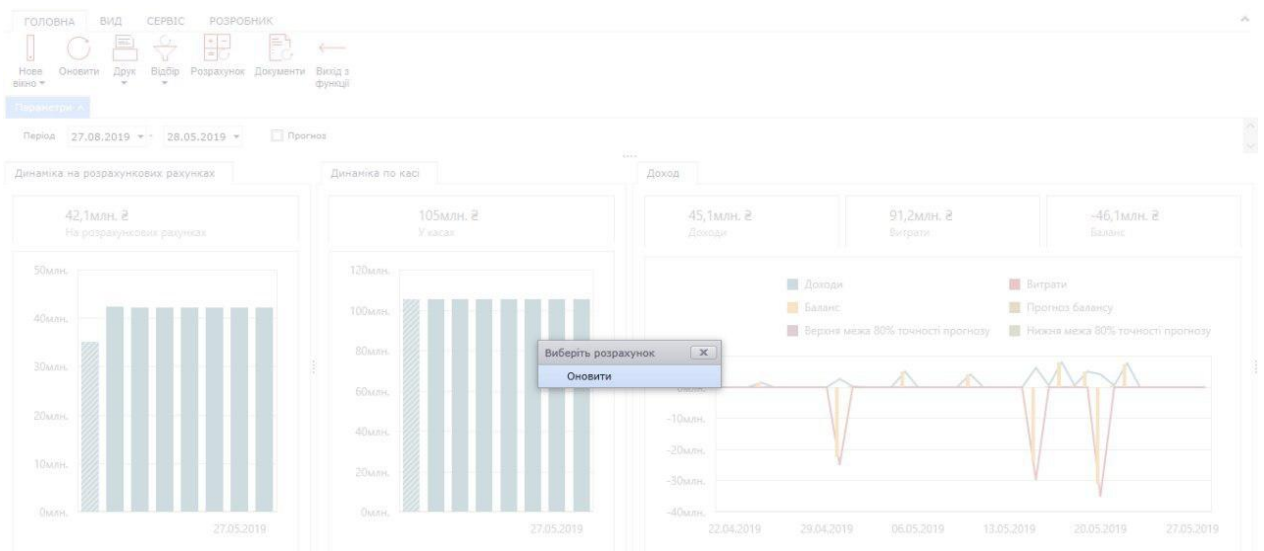


Рисунок 5.2 – Спосіб оновлення даних для відображення

При необхідності, керівник має змогу окремо переглянути кожну закладку, клікнувши по назві закладки двічі. Наводячи курсором на графік ста-

ну кожного фінансового показнику, можна побачити стан на конкретний день. Детальний перегляд закладок зображений на рисунка 5.3 – 5.5.

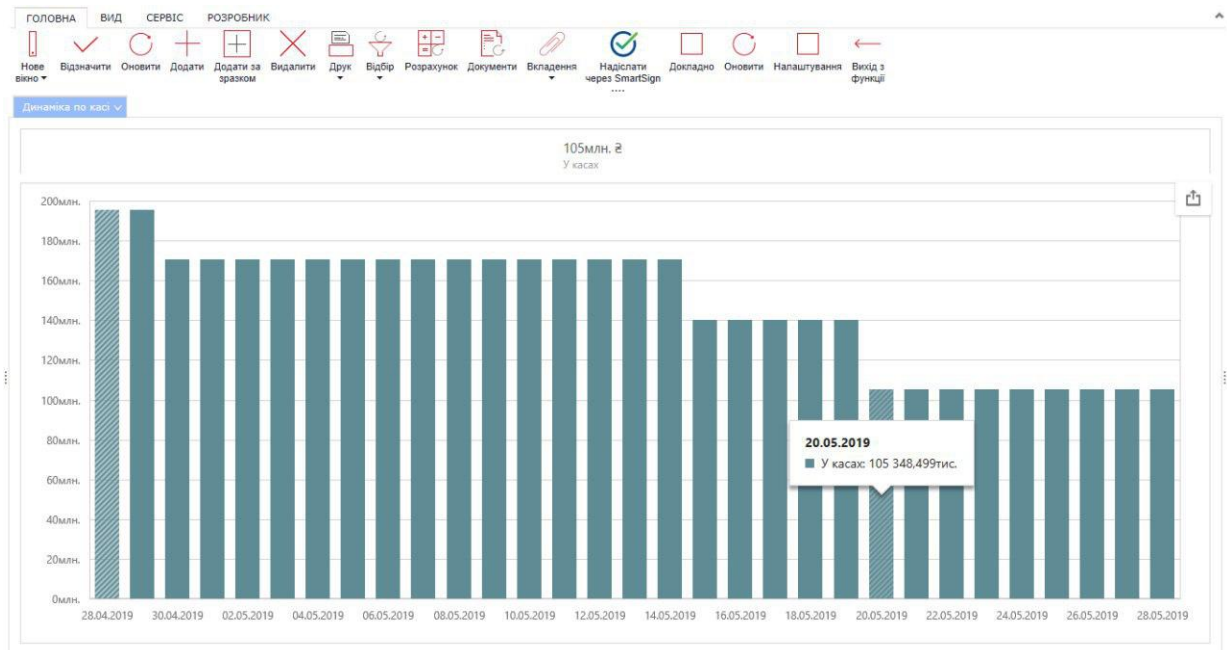


Рисунок 5.3 – Детальний перегляд закладки «Динаміка по касі»

Зверху, над графіком стану грошових коштів в касі, надана інформація про загальну суму коштів в касі на заданий період.

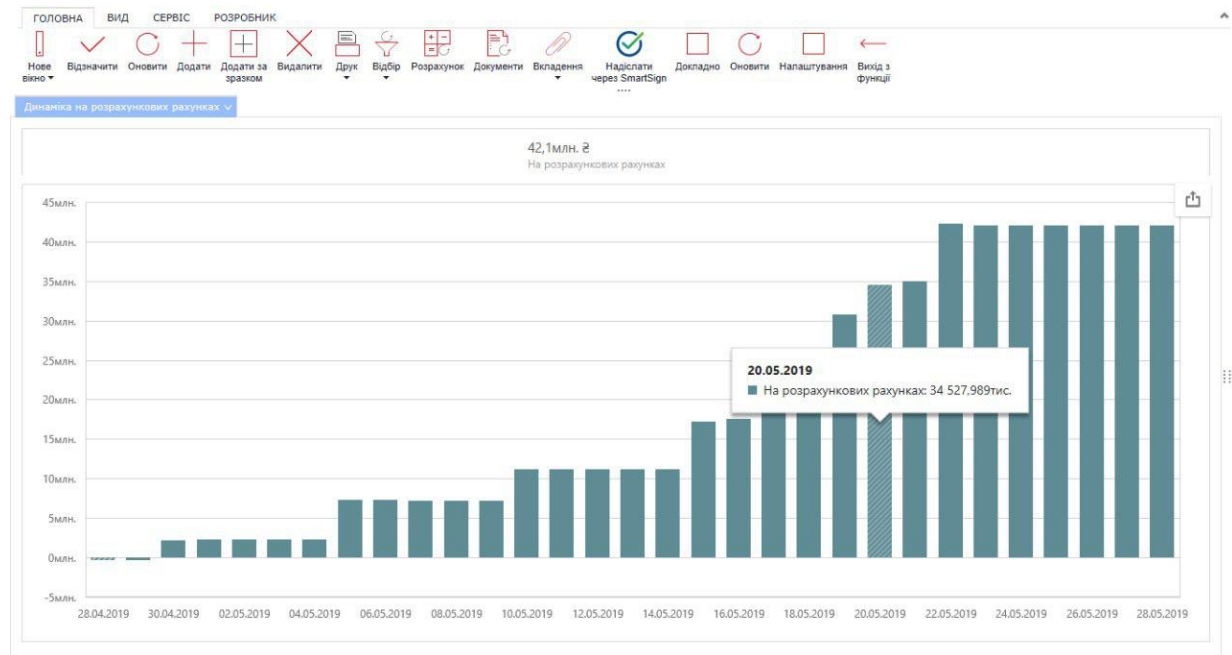


Рисунок 5.4 – Детальний перегляд закладки
«Динаміка на розрахункових рахунках»

Над графіком стану коштів на розрахункових рахунках в банках відображається загальна сума коштів на розрахункових рахунках за заданий період.

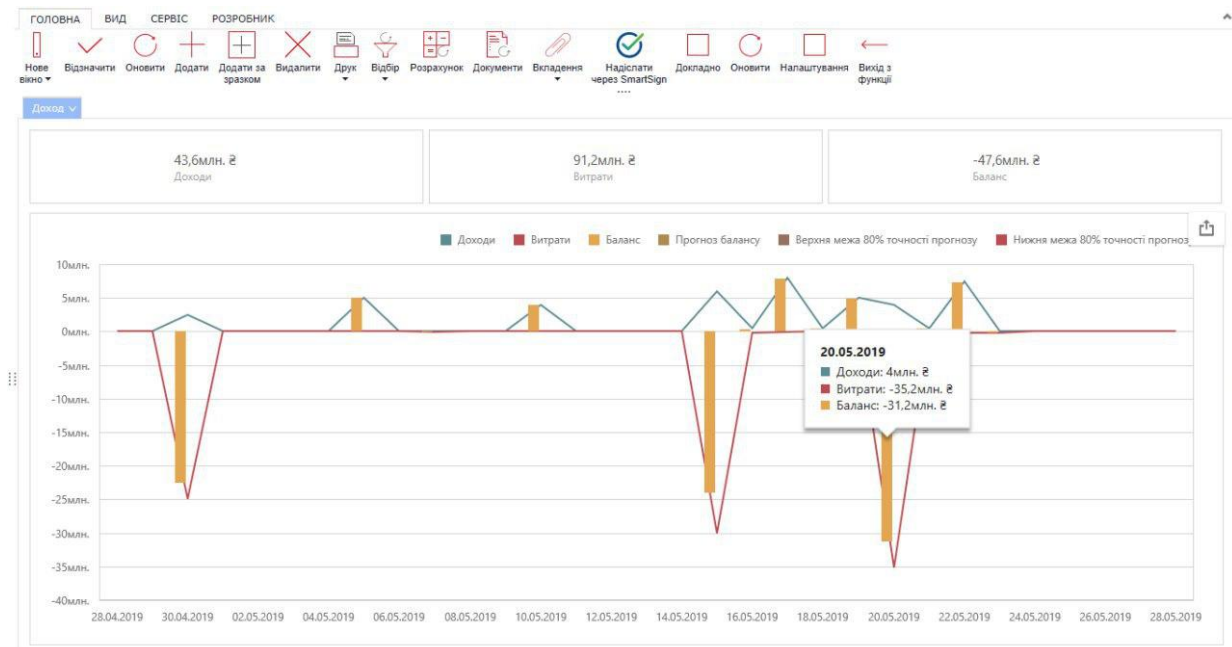


Рисунок 5.5 – Детальний перегляд закладки «Дохід»

Над відображенням стану балансу підприємства відображаються загальні доходи та витрати підприємства за заданий період, а також баланс підприємства, що являє собою різницю між загальними сумами доходів та витрат.

Для того, щоб керівник міг переглянути прогноз фінансових показників, йому необхідно натиснути галочку «Прогноз» та у полі, що з'явиться, ввести дату, до якої треба сформувати прогноз та виконати оновлення натиснувши кнопку «Розрахунок» та вибравши «Оновити». Процедура проведення прогнозування представлена на рисунку 5.6.

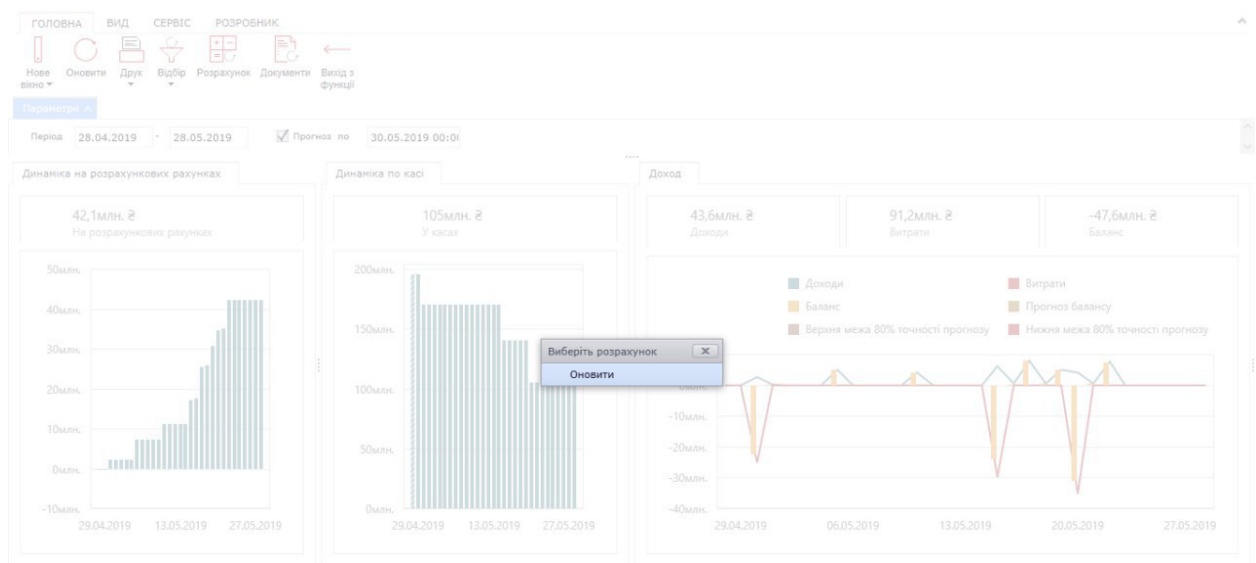


Рисунок 5.6 – Проведення процедури прогнозування

Після оновлення даних, для перегляду буде доступний прогноз, який на графіках станів фінансових показників відображатиметься іншим кольором, що представлено на рисунку 5.7.

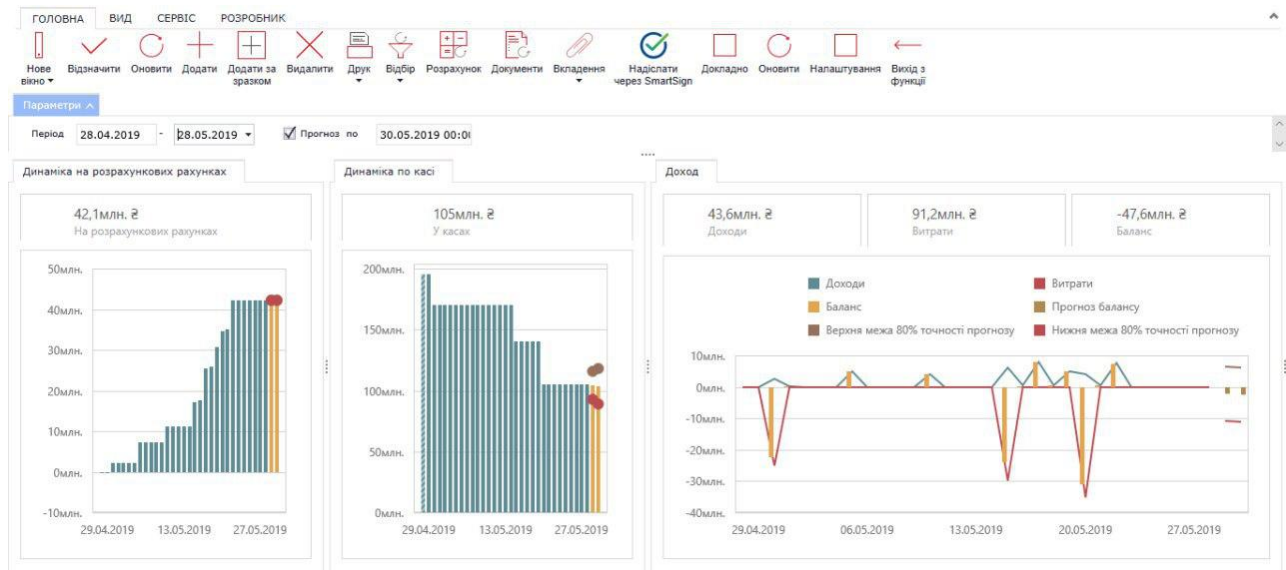


Рисунок 5.7 – Проведення процедури прогнозування

Як і під час перегляду станів грошових коштів в касі, станів коштів на розрахункових рахунках в банках та стану балансу підприємства, є можливість і під час прогнозування переглянути детальніше кожну закладку окремо. А наведення курсору на конкретний день, відобразить більше інформації про прогноз.

Для того, щоб переглянути оборотно-сальдову відомість, тобто залишки на початок і на кінець періоду, та обороти по дебету та по кредиту за вказаний період для кожного рахунку, керівнику необхідно стати на закладку того фінансового показнику, оборотно-сальдова відомість якого потрібна та натиснути кнопку «Докладно». У вікні, що відкриється, необхідно натиснути кнопку «Сформувати». Перегляд оборотно-сальдової відомості наведено на рисунку 5.8.

ГОЛОВНА

ВІД

СЕРВІС

РОЗРОБНИК

Нове вікно

Відзначити

Оновити

Друк

Відбір

Розрахунок

Документи

Аналітика

Сформувати

Вихід з цієї

Сформувати (Ctrl+Enter)

Параметри

Дата з 22.05.2019 по 28.05.2019

Рахунок 31

Профіль розрахунок

Організація

Обороти

Розгорнути до документів

Групувати

По днях

Підрозділи

Без кор. рахунків

Кор.рахунки

Тільки детальні рядки

Показники

Рахунки

Обороти

Форма

Спеціальні

ПрАТ "Літак" + Додати

Оборотно-сальдова відомість

Звіт

Контрагент	P/r	Період	Де	Вал	Сальдо	Дебет	Кредит	Сальдо	
				ОВ	початкове			кінцеве	
▲ (1) Борисполь ТзОВ "Демо"			✓	Н	ГРН	Д 35 029 988.50	7 500 000.00	450 000.00	Д 42 079 988.50
▲ (1) Борисполь ТзОВ "Демо"	(1) Банк: 380805 p/c:1234567890		✓	Н	ГРН	К 1 237 410.80			К 1 237 410.80
▲ (1) Борисполь ТзОВ "Демо"	(2) Банк: 325365 p/c:260000000001		✓	Н	ГРН	К 1 124 189.24			К 1 124 189.24
▲ (1) Борисполь ТзОВ "Демо"	(5) Банк: 334851 p/c:26000000000003		✓	Н	ГРН	К 749 773.79			К 749 773.79
▲ (1) Борисполь ТзОВ "Демо"	(12) Банк: 380805 p/c:2600576591		✓	Н	ГРН	Д 633 505.52			Д 633 505.52
▲ (1) Борисполь ТзОВ "Демо"	(25) Банк: 325365 p/c:260000000001		✓	Н	ГРН	Д 4 539.30			Д 4 539.30
▲ (12) Київ ПАО "Банк"			✓	Н	ГРН	К 1 492.59			К 1 492.59
▲ (12) Київ ПАО "Банк"	() <не вказано>		✓	Н	ГРН	К 500.00			К 500.00
▲ (13180) Жашков ПрАТ "Літак"			✓	Н	ГРН	К 500.00			К 500.00
▲ (13180) Жашков ПрАТ "Літак"	(6) Банк: 334851 p/c:2600124578954		✓	Н	ГРН	Д 35 624 522.61	7 500 000.00	450 000.00	Д 42 674 522.61
▲ (13180) Жашков ПрАТ "Літак"	(1) Банк: 303440 p/c:260000066666		✓	Н	ГРН	К 1 000.00			К 1 000.00
▲ (13180) Жашков ПрАТ "Літак"	(2) Банк: 303440 p/c:35792050070296		✓	Н	ГРН	К 8 389 671.42			К 8 389 671.42
▲ (13180) Жашков ПрАТ "Літак"	(3) Банк: 303440 p/c:260400066666		✓	Н	ГРН	Д 30 002.60			Д 30 002.60
▲ (13180) Жашков ПрАТ "Літак"	(4) Банк: 320478 p/c:26004212003729		✓	Н	ГРН	К 1 078.10			К 1 078.10
▲ (13180) Жашков ПрАТ "Літак"	(6) Банк: 303440 p/c:260000066666		✓	Н	ГРН	К 2 000.00			К 2 000.00
▲ (13180) Жашков ПрАТ "Літак"	(8) Банк: 351005 p/c:26002500055555		✓	Н	ГРН	Д 7 149 321.09			Д 7 149 321.09
▲ (13180) Жашков ПрАТ "Літак"	(8) Банк: 351005 p/c:26002500055555	Обороти за період: 22.05.2019	✓	Н	ГРН	Д 36 838 948.44	7 500 000.00	450 000.00	Д 43 888 948.44
▲ (13180) Жашков ПрАТ "Літак"	(8) Банк: 351005 p/c:26002500055555	Обороти за період: 23.05.2019	✓	Н	ГРН	Д 36 838 948.44	7 500 000.00	250 000.00	Д 44 088 948.44
▲ (13180) Жашков ПрАТ "Літак"	(8) Банк: 351005 p/c:26002500055555		✓	Н	ГРН	Д 44 088 948.44		200 000.00	Д 43 888 948.44
▲ () <не вказано>			✓	Н	ГРН	Д 643 376.69			Д 643 376.69
▲ () <не вказано>			✓	Н	ГРН	Д 643 376.69			Д 643 376.69

Рисунок 5.8 – Відображення оборотно-сальдової відомості показнику
«Динаміка по касі»

Креслення вигляду екранних форм наведені в графічному матеріалі.

5.2 Випробування програмного продукту

В підрозділі випробування програмного продукту наведено опис та порядок виконання тестів задля перевірки відповідності розробленого функціоналу комплексу задач функціональним вимогам, наведених у технічному завданні на оптимізацію продукту MASTER функціоналом для моніторингу фінансових показників.

5.2.1 Мета випробувань

Метою випробувань є перевірка відповідності функцій комплексу задач з моніторингу фінансових показників вимогам технічного завдання.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

					ДП ІС-5121.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

Під час процесу тестування було перевірено основну функціональність функціоналу «Монітор керівника».

У таблицях 5.1 – 5.5 було наведено випробування основних функціональних можливостей, згідно з діаграмою використання.

Таблиця 5.1 – Тести введення періоду моніторингу

Функція/ Use Case	Введення періоду моніторингу	
Дія:	Очікуваний результат:	Результат тесту:
Передумова		
Зайдіть до інтерфейсу «Монітор керівника»	Відкритий інтерфейс «Монітор керівника»	пройдений
Кроки тесту:		
Заповніть поля на закладці «Параметри» Дата початку періоду: 28.04.2019 Дата закінчення періоду: 28.05.2019 Галочка «Прогноз» не стоїть	Поля успішно заповнені	пройдений

Таблиця 5.2 – Тести перегляду графіків фінансових показників

Функція/ Use Case	Перегляд графіків фінансових показників	
Дія:	Очікуваний результат:	Результат тесту:
Передумова		
Зайдіть до інтерфейсу «Монітор керівника»	Відкритий інтерфейс «Монітор керівника»	пройдений
Кроки тесту:		
Заповніть поля на закладці «Параметри» Дата початку періоду: 28.04.2019 Дата закінчення періоду: 28.05.2019 Галочка «Прогноз» не стоїть	Поля успішно заповнені	пройдений
Натисніть кнопку «Розрахунок»	З'явився список можливих розрахунків	пройдений
Оберіть «Оновити»	Графіки станів усіх фінансових показників успішно оновились	пройдений

Таблиця 5.3 – Тести введення періоду прогнозування

Функція/ Use Case	Введення періоду для прогнозування	
Дія:	Очікуваний результат:	Результат тесту:
Передумова		
Зайдіть до інтерфейсу «Монітор керівника»	Відкритий інтерфейс «Монітор керівника»	пройдений
Кроки тесту:		
Заповніть поля на закладці «Параметри» Дата початку періоду: 28.04.2019 Дата закінчення періоду: 28.05.2019 Галочка «Прогноз» стоїть Поле дати закінчення прогнозу: 30.05.2019	Поля успішно заповнені	пройдений

Таблиця 5.4 – Тести перегляду спрогнозованих даних

Функція/ Use Case	Перегляд спрогнозованих даних	
Дія:	Очікуваний результат:	Результат тесту:
Передумова		
Зайдіть до інтерфейсу «Монітор керівника»	Відкритий інтерфейс «Монітор керівника»	пройдений
Кроки тесту:		
Заповніть поля на заклад- ці «Параметри» Дата початку періоду: 28.04.2019 Дата закінчення періоду: 28.05.2019 Галочка «Прогноз» стоїть Поле дати закінчення прогнозу: 30.05.2019	Поля успішно заповнені	пройдений
Натисніть кнопку «Розра- хунок»	З'явився список можливих роз- рахунків	пройдений
Оберіть «Оновити»	Графіки станів усіх фінансових показників успішно оновились. Прогнози відображаються ко- льором відмінним від кольору графіка фактичного стану	пройдений

Таблиця 5.5 – Тести переходу до оборотно-сальдової відомості

Функція/ Use Case	Перехід до оборотно-сальдової відомості	
Дія:	Очікуваний результат:	Результат тесту:
Передумова		
Зайдіть до інтерфейсу «Монітор керівника»	Відкритий інтерфейс «Монітор керівника»	пройдений
Заповніть поля на закладці «Параметри» Дата початку періоду: 28.04.2019 Дата закінчення періоду: 28.05.2019 Галочка «Прогноз» не стоїть	Поля успішно заповнені	пройдений
Натисніть кнопку «Розрахунок»	З'явився список можливих розрахунків	пройдений
Оберіть «Оновити»	Графіки станів усіх фінансових показників успішно оновились	пройдений
Кроки тесту:		
Клікніть на закладку «Динаміка на розрахункових рахунках»	Заголовок закладки «Динаміка на розрахункових рахунках» став синім	пройдений
Натисніть кнопку «Докладно»	Відкрилось вікно оборотно-сальдової відомості	пройдений

Продовження таблиці 5.5

Натисніть кнопку «Сформува- ти»	Оборотно-сальдова ві- домість сформувалась	пройдений
Післяумова		
Натисніть на кнопку «Вихід з функції»	Вихід виконано. Відк- рито інтерфейс «Мо- нітор керівника»	пройдений

Висновок до розділу

У цьому розділі було описано інструкції для керівника. Детально було описано, як змінювати період моніторингу та період для прогнозування, як оновлювати дані, щоб стан фінансових показників відображався за вказаний період та як переходити до оборотно-сальдової відомості.

Було наведено мету випробувань та загальні положення для функціо-
налу «Монітор керівника» як засобу моніторингу фінансових показників.

Було розроблено тестові сценарії згідно з сценаріями діаграми викори-
стання та успішно пройдено їх.

ЗАГАЛЬНІ ВИСНОВКИ

У дипломній роботі було розглянуто основні задачі для оптимізації продукту MASTER шляхом розробки функціоналу «Монітор керівника», який надає можливість моніторингу фінансових показників.

У розділ загальних положень було надано інформації про предметне середовище роботи. Також було описано єдиного актора системи – керівника та його функції. Було проведено огляд наявних аналогів, який лише підкреслив актуальність та доцільність створення функціоналу «Монітор керівника».

У розділі інформаційного забезпечення було детально описано структуру сховища даних, вхідних та вихідних даних.

У розділі програмної та технічної підтримки було описано технології, які використовувались при розробці функціоналу моніторингу фінансових показників. В якості бази даних було обрано MsSql, інструментом розробки було обрано мову програмування C#, а засобом відображення станів фінансових показників було обрано DevExpress Dashboards. Для нормальної роботи функціоналу, було наведено вимоги до технічного та програмного забезпечень.

У технологічному розділі було наведено керівництво користувача, яке містить детальні інструкції керівнику про те, як користуватись розробленим функціоналом. Також було описано методи випробування на основі тестових сценаріїв.

Розроблений функціонал «Монітор керівника» вдосконалив продукт MASTER, додавши до нього засіб для моніторингу фінансових показників таких, як стан грошових коштів в касі, стан коштів на розрахункових рахунках в банках та баланс підприємства.

ПЕРЕЛІК ПОСИЛАНЬ

1. MASTER: Бюджетні установи [Електронний ресурс]. - Режим доступу: <https://masterbuh.com/product/3> - Дата доступу: 28.03.2019
2. Облік по-українськи – як працює вітчизняна альтернатива «1С» [Електронний ресурс]. - Режим доступу: <https://nachasi.com/2017/11/21/mb-alternatyva-1s/> - Дата доступу: 21.11.2017
3. Оборотно-сальдовая ведомость [Електронний ресурс]. - Режим доступу: https://ru.wikipedia.org/wiki/Оборотно-сальдовая_ведомость - Дата доступу: 29.10.2018
4. SAP SE [Електронний ресурс]. - Режим доступу: https://uk.wikipedia.org/wiki/SAP_SE - Дата доступу: 16.02.2019.
5. 1С: Бухгалтерія для України [Електронний ресурс]. - Режим доступу: https://uk.wikipedia.org/wiki/1С:_Бухгалтерія_для_України - Дата доступу: 14.01.201
6. Указ президента України №133/2017 [Електронний ресурс]. - Режим доступу: <https://www.president.gov.ua/documents/1332017-21850> - Дата доступу: 28.04.2017
7. Novo Forecast Enterprise [Електронний ресурс]. - Режим доступу: <https://novoforecast.com/novo-forecast-enterprise.html> - Дата доступу: 16.05.2019
8. Функции планирования. Прогнозирование [Електронний ресурс]. - Режим доступу: <https://ompartners.com/ru-RU/planning-functions/forecasting> - Дата доступу: 29.05.2019
9. Прогноз по методу экспоненциального сглаживания с трендом и сезонностью Хольта-Винерса [Електронний ресурс]. - Режим доступу: <https://4analytics.ru/prognozirovanie/prognoz-po-metodu-eksponencialnogo->

sglajivaniya-s-trendom-i-sezonnostyu-xolta-vintersa.html - Дата доступу:
12.07.2018

10. UML Class Diagram Tutorial [Електронний ресурс]. - Режим доступу:
<https://www.lucidchart.com/pages/uml-class-diagram> - Дата доступу:
20.04.2019
11. UML Sequence Diagram Tutorial [Електронний ресурс]. - Режим
доступу: <https://www.lucidchart.com/pages/uml-sequence-diagram> - Дата
доступу: 20.04.2019
12. Component Diagram Tutorial [Електронний ресурс]. - Режим доступу:
<https://www.lucidchart.com/pages/uml-component-diagram> - Дата доступу:
20.04.2019
13. Введение в C# [Електронний ресурс]. - Режим доступу:
<https://metanit.com/sharp/tutorial/1.1.php> – Дата доступу: 09.03.2017
14. Что такое SQL Server и T-SQL [Електронний ресурс]. - Режим доступу:
<https://metanit.com/sql/sqlserver/1.1.php> – Дата доступу: 09.03.2017
15. DevExpress Dashboard Architecture [Електронний ресурс]. – Режим
доступу: <https://documentation.devexpress.com/Dashboard/116677/Basic-Concepts-and-Terminology/DevExpress-Dashboard-Architecture> – Дата
доступу: 14.11.2018
16. DevExpress (Develpoer Express Inc) [Електронний ресурс]. - Режим
доступу: <http://www.interface.ru/home.asp?artId=29317&vId=364> – Дата
доступу: 27.05.2019

Додаток А

Тексти програмного коду*Комплекс задач з моніторингу фінансових показників бюджетних організацій*

(Найменування програми (документа))

DVD-R

(Вид носія даних)

7 арк, 244 Кб

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2019 року

					ДП ІС-5121.1153-с.ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		


```

public class Forecast
{
    private string _mainDataSource = string.Empty;
    private bool _lastDataEqualsForecast = true;
    private DateTime _dateNow = Settings.DateNow;

    private DateAggregatorEtsForecastResult forecastCreating(DateTime endDate, DateTime beginDate, List<Tuple<DateTime, double>> data)
    {
        if(endDate <= _dateNow || beginDate > _dateNow)
        {
            return null;
        }
        var forecastList = new List<DateValue<double>>();
        foreach (var d in data.Where(x => x.Item1 <= _dateNow))
        {
            forecastList.Add(new DateValue<double>(d.Item2, d.Item1));
        }
        var forecastLength = (endDate - _dateNow).Days;
        var requestFactory = ServiceLocator.Get<IDateAggregatorEtsForecastRequestFactory>();
        var request = requestFactory.Create(forecastList.Select(x => new DateValue<double>(x.Value, x.Date)), forecastLength, "AAN", fillEmptyValuesMethod: FillEmptyValuesMethod.BackFill);
        var forecastService = ServiceLocator.Get<IDateAggregatorEtsForecastService>();
        var result = forecastService.Ets(request);
        return result;
    }

    public void Create(string mainDataSource, string sign, bool lastDataEqualsForecast = true)
    {
        _mainDataSource = mainDataSource;
        _lastDataEqualsForecast = lastDataEqualsForecast;

        var dates = SqlClient.Local.CreateCommand(string.Format("select min(CFLDS004_DDM) as beginDate, max(CFLDS004_DDM) as endDate from {0}", mainDataSource)).ExecObject(new { beginDate = DateTime.MinValue, endDate = DateTime.MinValue });

        if (Text.Compare(sign, "Bank") || Text.Compare(sign, "Cash"))
        {
            var data = SqlClient.Local.CreateCommand(string.Format("select CFLDS004_SUMDB as summa, CFLDS004_DDM as dateDdm from {0} where CFLDS004_DDM between @begin and @end", mainDataSource),
                new SqlParameter("begin", dates.beginDate),
                new SqlParameter("end", dates.endDate)).ExecObjects(new { dateDdm = DateTime.MinValue, summa = double.MinValue }).Select(x => Tuple.Create(x.dateDdm, x.summa)).ToList();

            var forecast = forecastCreating(dates.endDate, dates.beginDate, data);
            bankCashForecastUpdate(forecast, data);
        }
        if (Text.Compare(sign, "Balance"))
        {
            var data = SqlClient.Local.CreateCommand(string.Format("select sum(CFLDS004_ENTDB) as income, sum(CFLDS004_ENTCR) as consumption, CFLDS004_DDM as dateDdm from {0} where CFLDS004_DDM between @begin and @end group by CFLDS004_DDM", mainDataSource),
                new SqlParameter("begin", dates.beginDate),
                new SqlParameter("end", dates.endDate)).ExecObjects(new { income = double.MinValue, consumption = double.MinValue, dateDdm = DateTime.MinValue }).Select(x => Tuple.Create(x.dateDdm, x.income, x.consumption)).ToList();

            var incomeData = new List<Tuple<DateTime, double>>();
            var consumptionData = new List<Tuple<DateTime, double>>();
            foreach (var d in data.Select(x => Tuple.Create(x.Item1, x.Item2)))
            {
                incomeData.Add(Tuple.Create(d.Item1, d.Item2)); // доход
            }
            var incomeForecast = forecastCreating(dates.endDate, dates.beginDate, incomeData);
            balanceForecastUpdate(incomeForecast, true, incomeData);
        }
    }
}

```

```

x.Item3)))
        foreach (var d in data.Select(x => Tuple.Create(x.Item1,
        {
            consumptionData.Add(Tuple.Create(d.Item1, d.Item2));
        }
        var consumptionForecast = forecastCreating(dates.endDate,
        dates.beginDate, consumptionData);
        balanceForecastUpdate(consumptionForecast, false, consumptionDa-
        ta);
        }
        if(Text.Compare(sign, "Buy"))
        {
            var data = SqlClient.Local.CreateCommand(string.Format("select
            sum(CFLDS004_ENTDB) as buy, CFLDS004_DDM as dateDdm from {0} where CFLDS004_DDM between @begin and @end
            group by CFLDS004_DDM", _mainDataSource),
            new SqlParameter("begin", dates.beginDate),
            new SqlParameter("end", dates.endDate)).ExecObjects(new { da-
            teDdm = DateTime.MinValue, buy = double.MinValue }).Select(x => Tuple.Create(x.dateDdm, x.buy)).ToList();
            var forecast = forecastCreating(dates.endDate, dates.beginDate,
            data);
            balanceForecastUpdate(forecast, true, data);
        }
        if (Text.Compare(sign, "Mbuy"))
        {
            var data = SqlClient.Local.CreateCommand(string.Format("select
            sum(CFLDS004_ENTCR) as sales, CFLDS004_DDM as dateDdm from {0} where CFLDS004_DDM between @begin and @end
            group by CFLDS004_DDM", _mainDataSource),
            new SqlParameter("begin", dates.beginDate),
            new SqlParameter("end", dates.endDate)).ExecObjects(new { da-
            teDdm = DateTime.MinValue, sales = double.MinValue }).Select(x => Tuple.Create(x.dateDdm,
            x.sales)).ToList();
            var forecast = forecastCreating(dates.endDate, dates.beginDate,
            data);
            balanceForecastUpdate(forecast, false, data);
        }
        if (Text.Compare(sign, "Sales"))
        {
            var data = SqlClient.Local.CreateCommand(string.Format("select
            sum(CFLDS004_ENTCR) as sales, CFLDS004_DDM as dateDdm from {0} where CFLDS004_DDM between @begin and @end
            group by CFLDS004_DDM", _mainDataSource),
            new SqlParameter("begin", dates.beginDate),
            new SqlParameter("end", dates.endDate)).ExecObjects(new { da-
            teDdm = DateTime.MinValue, sales = double.MinValue }).Select(x => Tuple.Create(x.dateDdm,
            x.sales)).ToList();
            var forecast = forecastCreating(dates.endDate, dates.beginDate,
            data);
            balanceForecastUpdate(forecast, false, data);
        }
        if (Text.Compare(sign, "Msale"))
        {
            var data = SqlClient.Local.CreateCommand(string.Format("select
            sum(CFLDS004_ENTDB) as buy, CFLDS004_DDM as dateDdm from {0} where CFLDS004_DDM between @begin and @end
            group by CFLDS004_DDM", _mainDataSource),
            new SqlParameter("begin", dates.beginDate),
            new SqlParameter("end", dates.endDate)).ExecObjects(new { da-
            teDdm = DateTime.MinValue, buy = double.MinValue }).Select(x => Tuple.Create(x.dateDdm, x.buy)).ToList();
            var forecast = forecastCreating(dates.endDate, dates.beginDate,
            data);
            balanceForecastUpdate(forecast, true, data);
        }
        }
        private void merge(List<Tuple<DateTime, double>> data, string sign, bool isIncome
        = false)
        {
            if (data == null || !data.Any())
            {
                return;
            }
            var summ = data.Where(x => x.Item1 == _dateNow).Select(y =>
            y.Item2).FirstOrDefault();
            var isBankCash = Text.Compare(sign, "BankCash");
            var updateBuilder = SqlClient.Local.CreateUpdateBuilder();
            updateBuilder.Table.Name = _mainDataSource;
            if (_lastDataEqualsForecast)
            {
                updateBuilder.Set.SetValue(isBankCash ? "CFLDS004_FCAST" : (isIn-
                come ? "CFLDS004_CASTI" : "CFLDS004_CASTC"), summ);
            }
        }
    }
}

```

```

        updateBuilder.Set.SetValue(isBankCash ? "CFLDS004_L080" : (isIncome ? "CFLDS004_L080I" : "CFLDS004_L080C"), summ);
        updateBuilder.Set.SetValue(isBankCash ? "CFLDS004_HI80" : (isIncome ? "CFLDS004_HI80I" : "CFLDS004_HI80C"), summ);
    }
    else
    {
        updateBuilder.Set.SetExpression(isBankCash ? "CFLDS004_FCAST" : (isIncome ? "CFLDS004_CASTI" : "CFLDS004_CASTC"), "null");
        updateBuilder.Set.SetExpression(isBankCash ? "CFLDS004_L080" : (isIncome ? "CFLDS004_L080I" : "CFLDS004_L080C"), "null");
        updateBuilder.Set.SetExpression(isBankCash ? "CFLDS004_HI80" : (isIncome ? "CFLDS004_HI80I" : "CFLDS004_HI80C"), "null");
    }
    updateBuilder.Where = new SqlCommandText("CFLDS004_DDM = @ddm", new SqlParameter("ddm", _dateNow));
    updateBuilder.Exec();
}

private void bankCashForecastUpdate(DateAggregatorEtsForecastResult forecast, List<Tuple<DateTime, double>> data)
{
    if (forecast != null && forecast.Forecast != null && forecast.Forecast.Any())
    {
        double lastSumm = 0;
        merge(data, "BankCash");
        var forecastInfo = forecast.Forecast;
        if(forecastInfo != null && forecastInfo.Any())
        {
            var updateBuilder = SqlClient.Local.CreateUpdateBuilder();
            updateBuilder.Table.Name = _mainDataSource;
            foreach(var f in forecastInfo)
            {
                lastSumm = f.Value.Forecast;
                updateBuilder.Set.SetExpression("CFLDS004_SUMDB", "null");
                updateBuilder.Set.SetValue("CFLDS004_FCAST", lastSumm);
                updateBuilder.Set.SetValue("CFLDS004_L080", f.Value.Lo80);
                updateBuilder.Set.SetValue("CFLDS004_HI80", f.Value.Hi80);
                updateBuilder.Where = new SqlCommandText("CFLDS004_DDM = @ddm", new SqlParameter("ddm", f.Date));
                updateBuilder.Exec();
            }
        }
        var summ = SqlClient.Local.CreateCommand(string.Format("select CFLDSUMNDB from {0} where SUM_DATE<=@ddm order by SUM_DATE DESC", _mainDataSource), new SqlParameter("ddm", _dateNow)).ExecScalar<decimal>();
        SqlClient.Local.CreateCommand(string.Format("@update {0} set CFLDS004_SUML = @summ", _mainDataSource), new SqlParameter("summ", summ)).ExecNonQuery();
    }
}

private void balanceForecastUpdate(DateAggregatorEtsForecastResult forecast, bool isIncome, List<Tuple<DateTime, double>> data)
{
    if (forecast != null && forecast.Forecast != null && forecast.Forecast.Any())
    {
        merge(data, "Balance", isIncome);
        var forecastInfo = forecast.Forecast;
        if(forecastInfo != null && forecastInfo.Any())
        {
            var updateBuilder = SqlClient.Local.CreateUpdateBuilder();
            updateBuilder.Table.Name = _mainDataSource;
            foreach(var f in forecastInfo)
            {
                updateBuilder.Set.SetExpression(isIncome ? "CFLDS004_ENTDB" : "CFLDS004_ENTCR", "null");
                updateBuilder.Set.SetValue(isIncome ? "CFLDS004_CASTI" : "CFLDS004_CASTC", f.Value.Forecast);
            }
        }
    }
}

```

```

        "CFLDS004_L080I" : "CFLDS004_L080C", f.Value.Lo80);
        "CFLDS004_HI80I" : "CFLDS004_HI80C", f.Value.Hi80);
        = @ddm", new SqlParameter("ddm", f.Date));
    }
}

public DateAggregatorEtsForecastRequestFactory(ITimeSeriesPreprocessorService preprocessorService)
{
    PreprocessorService = preprocessorService;

    public ITimeSeriesPreprocessorService PreprocessorService { get; private set; }

    public DateAggregatorEtsForecastRequest Create(IEnumerable<DateValue<double>>
timeSerie, int forecastHorizont, string algorithm = "ZZZ", int seasonality = 0, DateAggregationMethod ag-
gregation = DateAggregationMethod.Day, FillEmptyValuesMethod fillEmptyValuesMethod = FillEmptyValuesMe-
thod.Zeros)
    {
        return new DateAggregatorEtsForecastRequest(PreprocessorService, timeSe-
rie, aggregation, fillEmptyValuesMethod, algorithm, forecastHorizont, seasonality);
    }

    public class DateAggregatorEtsForecastRequest : EtsForecastServiceRequest
    {
        public DateAggregatorEtsForecastRequest(ITimeSeriesPreprocessorService timeSerie-
PreprocessorService, IEnumerable<DateValue<double>> timeSerie, DateAggregationMethod aggregation, FillEmp-
tyValuesMethod fillEmptyValuesMethod, string algorithm, int forecastHorizont, int seasonality)
            : base(null, forecastHorizont, algorithm, seasonality)
        {
            TimeSeriesPreprocessorService = timeSeriePreprocessorService;
            TimeSerie = timeSerie;
            Aggregation = aggregation;
            FillEmptyValuesMethod = fillEmptyValuesMethod;
        }
        public virtual IEnumerable<DateValue<double>> TimeSerie { get; private set; }
        public virtual DateAggregationMethod Aggregation { get; private set; }
        public virtual FillEmptyValuesMethod FillEmptyValuesMethod { get; private set; }

        private ITimeSeriesPreprocessorService TimeSeriePreprocessorService { get; set; }

        public override IEnumerable<double> Serie
        {
            get
            {
                var result = TimeSeriePrepared;
                return result.Select(i=>i.Value);
            }
        }

        public virtual IEnumerable<DateValue<double>> TimeSeriePrepared
        {
            get
            {
                var result = TimeSeriePreprocessorSer-
vice.PreProcessTimeSerie(TimeSerie, Aggregation, FillEmptyValuesMethod);
                return result;
            }
        }
    }

    public class DateAggregatorEtsForecastResult : ServiceResult
    {
        public DateAggregatorEtsForecastResult(DateAggregatorEtsForecastRequest inputPa-
rams, IEnumerable<DateValue<EtsHiloForecast>> forecast, bool success, string message)
        {
            InputParams = inputParams;
            Forecast = forecast;
            Success = success;
            Message = message;
        }
        public virtual DateAggregatorEtsForecastRequest InputParams { get; private set; }
    }
}

```

```

        public virtual IEnumerable<DateValue<EtsHiLoForecast>> Forecast { get; private
set; }
    }

    public class DateValue<T>
    {
        public DateValue(T value, DateTime dateTime)
        {
            Value = value;
            Date = dateTime;
        }
        public virtual DateTime Date { get; private set; }
        public virtual T Value { get; private set; }
    }

    public class DateAggregatorEtsForecastRequest : EtsForecastServiceRequest
    {
        public DateAggregatorEtsForecastRequest(ITimeSeriesPreprocessorService timeSeries-
PreprocessorService, IEnumerable<DateValue<double>> timeSeries, DateAggregationMethod aggregation, FillEmp-
tyValuesMethod fillEmptyValuesMethod, string algorithm, int forecastHorizont, int seasonality)
            : base(null, forecastHorizont, algorithm, seasonality)
        {
            TimeSeriesPreprocessorService = timeSeriesPreprocessorService;
            TimeSeries = timeSeries;
            Aggregation = aggregation;
            FillEmptyValuesMethod = fillEmptyValuesMethod;
        }
        public virtual IEnumerable<DateValue<double>> TimeSeries { get; private set; }
        public virtual DateAggregationMethod Aggregation { get; private set; }
        public virtual FillEmptyValuesMethod FillEmptyValuesMethod { get; private set; }

        private ITimeSeriesPreprocessorService TimeSeriesPreprocessorService { get; set; }

        public override IEnumerable<double> Serie
        {
            get
            {
                var result = TimeSeriesPrepared;
                return result.Select(i=>i.Value);
            }
        }

        public virtual IEnumerable<DateValue<double>> TimeSeriesPrepared
        {
            get
            {
                var result = TimeSeriesPreprocessorSer-
vice.PreProcessTimeSeries(TimeSeries, Aggregation, FillEmptyValuesMethod);
                return result;
            }
        }
    }

    public class DateAggregatorEtsForecastResult : ServiceResult
    {
        public DateAggregatorEtsForecastResult(DateAggregatorEtsForecastRequest inputPa-
rams, IEnumerable<DateValue<EtsHiLoForecast>> forecast, bool success, string message)
        {
            InputParams = inputParams;
            Forecast = forecast;
            Success = success;
            Message = message;
        }
        public virtual DateAggregatorEtsForecastRequest InputParams { get; private set; }
        public virtual IEnumerable<DateValue<EtsHiLoForecast>> Forecast { get; private
set; }
    }

    public class DateValue<T>
    {
        public DateValue(T value, DateTime dateTime)
        {
            Value = value;
            Date = dateTime;
        }
        public virtual DateTime Date { get; private set; }
    }

```

```

        public virtual T Value { get; private set; }
    }

    public class LeaderdashEditor : DataEditorBusinessLogic.DataEditor
    {
        public override bool When()
        {
            if (!base.When())
            {
                return false;
            }

            var refresherPars = new ReportDataRefresherParams { Begin = Settings.DateNow };
            Dialog.Properties.Set(ReportDataRefresherParams.PropertyName, refresherPars);

            foreach (var c in Dialog.Cursors)
            {
                var bl = c.GetBusinessLogic<AccrepCursor>() as IReportConfigCursor;

                if (bl == null)
                {
                    continue;
                }
                var config = bl.Config;
                if (config == null)
                {
                    continue;
                }

                DateTime begin, end;
                config.GetPeriod(Settings.DateNow, out begin, out end);
                if (refresherPars.DateRun < end)
                {
                    refresherPars.DateRun = end;
                }

                if (refresherPars.Begin > begin)
                {
                    refresherPars.Begin = begin;
                }
            }

            Dialog.Cursors.Find("PARAMS").Frames.Get("PARAMS").FixedHeight = 1.3m;

            return true;
        }
    }

    public class Leadbank : DashboardLogicBase
    {
        public override List<DataSource> GetAdditionalDataSources()
        {
            RefreshData();
            return base.GetAdditionalDataSources();
        }

        public override void RefreshData()
        {
            new Forecast().Create(MainDataSource, "Bank", false);
            base.RefreshData();
        }
    }

    public class Leadcash : DashboardLogicBase
    {
        public override List<DataSource> GetAdditionalDataSources()
        {
            RefreshData();
            return base.GetAdditionalDataSources();
        }

        public override void RefreshData()
        {
            new Forecast().Create(MainDataSource, "Cash", false);
            base.RefreshData();
        }
    }

```

```
public class Leadbalanc : DashboardLogicBase
{
    public override List<DataSource> GetAdditionalDataSources()
    {
        RefreshData();
        return base.GetAdditionalDataSources();
    }

    public override void RefreshData()
    {
        new Forecast().Create(MainDataSource, "Balance", false);
        base.RefreshData();
    }
}
```

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації та управління

УЗГОДЖЕНО

Керівник проекту

_____ О.В. Гриша
(підпис) (ініціали, прізвище)

“16” травня 2019 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ О.А. Павлов
(підпис) (ініціали, прізвище)

“17” травня 2019 р.

Комплекс задач з моніторингу фінансових показників
бюджетної організації

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр *ДП ІС-5121.1181-с.ТЗ*

на 10 сторінках

Київ – 2019 року

ЗМІСТ

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	3
1.1 Повне найменування системи та її умовне позначення	3
1.2 Найменування організації-замовника та організацій-учасників робіт	3
1.3 Перелік документів, на підставі яких створюється система	3
1.4 Планові терміни початку і закінчення роботи зі створення системи	3
2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СЕРВІСУ	5
2.1 Призначення розробки	5
2.2 Цілі створення функціоналу	5
3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....	6
4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	7
4.1 Вимоги до функціональних характеристик	7
4.2 Вимоги до надійності	7
4.3 Вимоги до складу і параметрів технічних засобів	7
5 СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	9
6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	10
6.1 Види випробувань.....	10

					ДП ІС-5121.1181-с.ТЗ						
		Прізвище	Підпис								
Розроб.	Прокопенко І.В.				Комплекс задач з моніторингу фінансових показників бюджетних організацій	Літ.		Арк.		Аркушів	
								2	10		
Перевірив.	Гриша О.В.					КПІ ФІОТ кафедра АСОІУ гр. ІС-51					
Н. кон.	Москаленко Н.В.										
Затв.	Павлов О.А.										

ДП ІС-5121.1181-с.ТЗ

ДП ІС-5121.1181-с.ТЗ

Арк.

2

Змн.	Арк.	№ докум.	Підпис	Дата

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Повне найменування системи та її умовне позначення

Повне найменування комплексу задач: комплекс задач з моніторингу фінансових показників для бюджетних організацій.

Умовне позначення: «Монітор керівника».

1.2 Найменування організації-замовника та організацій-учасників робіт

Замовником є ТОВ «МАСТЕР:ГЛОБАЛ» (далі за текстом Замовник).
Адреса замовника: вул. Єлизавети Чавдар 5.

Розробник функціоналу — студентка групи ІС-51 кафедри автоматизованих систем обробки інформації та управління Національного технічного університету України "Київський політехнічний інститут ім. Ігоря Сікорського" Прокопенко Ірина Вікторівна.

1.3 Перелік документів, на підставі яких створюється система

При розробці системи і створення проектно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи;
- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий строк початку роботи по розробці комплексу задач — 20

						ДП ІС-5121.1181-с.ТЗ	Арк.
лютого	2019 року.						3
Змн.	Арк.	№ докум.	Підпис	Дата			

Плановий строк закінчення роботи по розробці комплексу задач — 03 червня 2019 року.

					ДП ІС-5121.1181-с.ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СЕРВІСУ

2.1 Призначення розробки

Функціонал призначений для вдосконалення продукту MASTER шляхом розроблення функціоналу «Монітор керівника» для можливості моніторингу та прогнозування таких фінансових показників, як стан грошових коштів в касі, стан коштів на розрахункових рахунках в банках, стан балансу підприємства, задля можливості передбачення змін економічної ситуації та для полегшення роботи керівника підприємства щодо планування кращого функціонування цього підприємства.

2.2 Цілі створення функціоналу

Метою розробки є пришвидшення процесу сприйняття інформації щодо змін станів фінансових показників, що дає змогу спрогнозувати стан фінансових показників та миттєво зреагувати на ці зміни за рахунок візуалізації даних.

					ДП ІС-5121.1181-с.ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Для користування даним функціоналом, користувач повинен мати придбаним продукт MASTER.

Об'єктом автоматизації є функціонал «Монітор керівника», який відображає стан фінансових показників та надає можливість створення прогнозу фінансових показників.

					ДП ІС-5121.1181-с.ТЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

Реалізований функціонал забезпечує швидке сприйняття інформації щодо змін станів фінансових показників, що дає змогу спрогнозувати стан фінансових показників та миттєво зреагувати на ці зміни за рахунок візуалізації даних.

Функціонал повинен забезпечувати можливість виконання перерахованих нижче функцій:

- введення періодів моніторингу та прогнозування;
- перегляд графіків фінансових показників та спрогнозованих даних;
- перехід до оборотно-сальдової відомості.

4.2 Вимоги до надійності

Функціонал повинен нормально функціонувати, не зважаючи на можливі дефекти, які можуть проявитись під час впровадження та експлуатації. Доопрацювання виявлених недоліків повинно відбуватись на етапі попередніх випробувань, під час впровадження та експлуатації.

Будь-які відхилення в роботі функціоналу не повинні вплинути на цілісність даних в інформаційному сховищі та на роботу продукту в цілому.

4.3 Вимоги до складу і параметрів технічних засобів

Для правильної роботи розробленого функціоналу до складу технічних засобів мають входити:

- має бути встановлене таке програмне забезпечення:

1) Операційна система вище Windows 7;

2) Куплений продукт MASTER або купити продукт уже з

розробленим функціоналом.

ДП ІС-5121.1181-с.ТЗ

					Арк.
					7
Змн.	Арк.	№ докум.	Підпис	Дата	

— комп'ютерна периферія, до складу якої входить:

- 1) Монітор;
- 2) Мишка або тачпад;
- 3) Клавіатура.

5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Основні етапи виконання робіт з розробки системи ведення наукової роботи.

№ п/п	Назва етапу роботи	Термін виконання етапу	Результат виконання
1.	Підготовка технічного завдання на розробку програмного продукту	20.02.2019	
2.	Розробка сценарію роботи	01.03.2019	
3.	Технічне проектування – функціональність, модулі, задачі, цілі тощо	05.03.2019	
4.	Узгодження з керівником інтерфейсу користувача	10.03.2019	
5.	Розробка інформаційного забезпечення	17.03.2019	
6.	Розробка програмного забезпечення	29.03.2019	
7.	Налагодження програми	01.05.2019	
8.	Тестування програми	25.05.2019	
9.	Здача готового програмного продукту замовника	27.05.2019	

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

6.1 Види випробувань

Для перевірки правильності роботи функціоналу потрібно провести функціональне тестування, в ході якого буде здійснено випробування функціональних характеристик «Монітору керівника».

Тестування відображення стану фінансових показників полягає у зміні періоду, за який потрібно показати стан.

Тестування прогнозування полягає у зміні періоду відображення стану фінансових показників за майбутній період. Така частина графіку має відображатись кольором відмінним від кольору графіку фактичного стану фінансових показників.

					ДП ІС-5121.1181-с.ТЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації та управління

УЗГОДЖЕНО

Керівник проекту

(підпис) О.В.Гриша
(ініціали, прізвище)

“13” травня 2019 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

(підпис) О.А.Павлов
(ініціали, прізвище)

“14” травня 2019 р.

Комплекс задач з моніторингу фінансових показників
бюджетної організації

ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ

Шифр ДП ІС-5121.1181-с.ПМВ

на 13 сторінках

Київ – 2019 року

ЗМІСТ

1	ОБ'ЄКТ ВИПРОБУВАННЯ	3
1.1	Найменування програми.....	3
1.2	Область застосування	3
1.3	Умовне позначення програми	3
2	МЕТА ВИПРОБУВАНЬ	4
3	ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ	5
3.1	Вимоги до функціональних характеристик	5
3.1.1	Вимоги до складу виконуваних функцій	5
4	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	6
5	СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ	7
6	МЕТОДИ ВИПРОБУВАНЬ	8

					ДП ІС-5121.1181-с.ПМВ			
<i>Зм.</i>	<i>Арк.</i>	<i>Прізвище</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Прокопенко І.В.</i>			<i>Комплекс задач з моніторингу фінансових показників бюджетних організацій</i>	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
							<i>2</i>	<i>13</i>
<i>Перевірив.</i>		<i>Гриша О.В.</i>				<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51</i>		
<i>Н. кон.</i>		<i>Москаленко Н.В.</i>						
<i>Затв.</i>		<i>Павлов О.В.</i>						

1 ОБ'ЄКТ ВИПРОБУВАННЯ

1.1 Найменування програми

Повне найменування комплексу задач: комплекс задач з моніторингу фінансових показників бюджетних організацій.

1.2 Область застосування

Функціонал призначений для вдосконалення продукту MASTER шляхом розроблення функціоналу «Монітор керівника» для можливості моніторингу та прогнозування таких фінансових показників, як стан грошових коштів в касі, стан коштів на розрахункових рахунках в банках, стан балансу підприємства, задля можливості передбачення змін економічної ситуації та для полегшення роботи керівника підприємства щодо планування кращого функціонування цього підприємства.

1.3 Умовне позначення програми

Умовне позначення: «Монітор керівника».

Змн.	Арк.	№ докум.	Підпис	Дата

2 МЕТА ВИПРОБУВАНЬ

Метою випробувань є перевірка відповідності функцій комплексу задач з моніторингу фінансових показників вимогам технічного завдання.

					ДП ІС-5121.1181-с.ПМВ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

3.1 Вимоги до функціональних характеристик

Функціональні вимоги:

- введення періоду моніторингу;
- перегляд графіків фінансових показників;
- введення періоду для прогнозування;
- перегляд спрогнозованих даних;
- перехід до оборотно-сальдової відомості.

3.1.1 Вимоги до складу виконуваних функцій

Функціонал повинен забезпечувати можливість виконання перерахованих нижче функцій:

- введення періодів моніторингу та прогнозування;
- перегляд графіків фінансових показників та спрогнозованих даних;
- перехід до оборотно-сальдової відомості.

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Програмна документація має складатися з керівництва користувача та вихідних текстів програмного коду.

					ДП ІС-5121.1181-с.ПМВ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

5 СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ

Етапи випробувань:

- ознайомчий;
- виконавчий.

На ознайомчому етапі проводиться:

- перевірка комплектності програмної документації;
- перевірка комплектності складу технічних і програмних засобів.

Під час виконавчого етапу проводиться:

- перевірка відповідності технічних характеристик системи;
- перевірка ступеню виконання вимог функціонального призначення системи.

Функції, що підлягають перевірці:

- введення періоду моніторингу;
- формування графіків фінансових показників;
- введення періоду прогнозування;
- формування графіків спрогнозованих даних;
- перехід до оборотно-сальдової відомості.

Змн.	Арк.	№ докум.	Підпис	Дата

6 МЕТОДИ ВИПРОБУВАНЬ

У процесі тестування була перевірена основна функціональність засобу для моніторингу та прогнозування фінансових показників. У таблицях 6.1 – 6.5 наведено перелік випробувань основних функціональних можливостей.

Таблиця 6.1 – Тести введення періоду моніторингу

Функція/ Use Case	Введення періоду моніторингу	
Дія:	Очікуваний результат:	Результат тесту:
Передумова		
Зайдіть до інтерфейсу «Монітор керівника»	Відкритий інтерфейс «Монітор керівника»	пройдений
Кроки тесту:		
Заповніть поля на закладці «Параметри» Дата початку періоду: 28.04.2019 Дата закінчення періоду: 28.05.2019 Галочка «Прогноз» не стоїть	Поля успішно заповнені	пройдений

Таблиця 6.2 – Тести формування графіків фінансових показників

Функція/ Use Case		Перегляд графіків фінансових показників	
Дія:		Очікуваний результат:	Результат тесту:
Передумова			
Зайдіть до інтерфейсу «Монітор керівника»		Відкритий інтерфейс «Монітор керівника»	пройдений
Кроки тесту:			
Заповніть поля на закладці «Параметри» Дата початку періоду: 28.04.2019 Дата закінчення періоду: 28.05.2019 Галочка «Прогноз» не стоїть		Поля успішно заповнені	пройдений
Натисніть кнопку «Розрахунок»		З'явився список можливих розрахунків	пройдений
Оберіть «Оновити»		Графіки станів усіх фінансових показників успішно оновились	пройдений

Таблиця 6.3 – Тести введення періоду моніторингу

Функція/ Use Case	Введення періоду для прогнозування	
Дія:	Очікуваний результат:	Результат тесту:
Передумова		
Зайдіть до інтерфейсу «Монітор керівника»	Відкритий інтерфейс «Монітор керівника»	пройдений
Кроки тесту:		
Заповніть поля на закладці «Параметри» Дата початку періоду: 28.04.2019 Дата закінчення періоду: 28.05.2019 Галочка «Прогноз» стоїть Поле дати закінчення прогнозу: 30.05.2019	Поля успішно заповнені	пройдений

Таблиця 6.4 – Тести перегляду спрогнозованих даних

Функція/ Use Case	Перегляд спрогнозованих даних	
Дія:	Очікуваний результат:	Результат тесту:
Передумова		
Зайдіть до інтерфейсу «Монітор керівника»	Відкритий інтерфейс «Мо- нітор керівника»	пройдений
Кроки тесту:		
Заповніть поля на закладці «Параметри» Дата початку періоду: 28.04.2019 Дата закінчення періоду: 28.05.2019 Галочка «Прогноз» стоїть Поле дати закінчення про- гнозу: 30.05.2019	Поля успішно заповнені	пройдений
Натисніть кнопку «Розра- хунок»	З'явився список можливих розрахунків	пройдений
Оберіть «Оновити»	Графіки станів усіх фінан- сових показників успішно оновились. Прогнози відо- бражаються кольором від- мінним від кольору графіка	пройдений

Таблиця 6.5 – Тести переходу до оборотно-сальдової відомості

Функція/ Use Case	Перехід до оборотно-сальдової відомості	
Дія:	Очікуваний результат:	Результат тесту:
Передумова		
Зайдіть до інтерфейсу «Монітор керівника»	Відкритий інтерфейс «Монітор керівника»	пройдений
Заповніть поля на закладці «Параметри» Дата початку періоду: 28.04.2019 Дата закінчення періоду: 28.05.2019 Галочка «Прогноз» не стоїть	Поля успішно заповнені	пройдений
Натисніть кнопку «Розрахунок»	З'явився список можливих розрахунків	пройдений
Оберіть «Оновити»	Графіки станів усіх фінансових показників успішно оновились	пройдений
Кроки тесту:		
Клікніть на закладку «Динаміка на розрахункових рахунках»	Заголовок закладки «Динаміка на розрахункових рахунках» став синім	пройдений

Продовження таблиці 6.5

Натисніть кнопку «Докладно»	Відкрилось вікно оборотно-сальдової відомості	пройдений
Натисніть кнопку «Сформува-ти»	Оборотно-сальдова відомість сформувалась	пройдений
Післяумова		
Натисніть на кнопку «Вихід з функції»	Вихід виконано. Відкрито інтерфейс «Монітор керівника»	пройдений

Змн.	Арк.	№ докум.	Підпис	Дата

Графічний матеріал до дипломного проекту

на тему: Комплекс задач з моніторингу фінансових показників
бюджетної організації

Київ – 2019 року